archiyou

ARCHIYOU UX/UI ANALYSIS REPORT

How to grow Archiyou as an open source design platform

February 9th 2026

1. EVALUATION ARCHIYOU 2025

We have been developing Archiyou organically for more than 3 years with *the mission to digitalize and democratize design and building knowledge*. In this period we - as a very small group of developers - have built the core technology and platform and tried a lot of things to engage users.

With the support of NLNET/NGI Zero Commons Fund we have been able to start a collaboration with our great UX/UI designer Wessel, who contributed a shift to a broader perspective with an emphasis on users, their goals and motivations.

The shift of perspective and way of work is characterized by:

- **From developer-focused to user-focused**
  This coincides nicely with the maturing of the technical side of the platform
- **From getting it to work technically to getting it to work for people**
  We have been talking a lot with user groups to understand what we need to add to fit their needs. We started doing continuous quickfixes to make that fit.
- **From early adopters to minimal viable community**
  By understanding individual users we could see ways how to make a community
- **Appropriate tech that serves**
  By understanding how the tech serves an open design community we gain a renewed focus on what is really important and maintainable in the long term

In what follows we'll go briefly through the various phases of our analysis, user research, strategy and redesigns.

1.1. USERS AND THEIR INTERESTS

Based on our registered users, the anonymized behaviors of users on our website and personal interactions we formulated the following user groups and with user interviews mapped out their interests, usage and views on Archiyou.

	CONSUMER	Beautiful designs and presentation Get product is most important: Maybe DIY DIY only if success is guaranteed Maybe hire PRO or use a DIY kit?
	DIY CRAFT MAKER	Nice designs and presentation for weekend project Good guarantee of successful builds: cost, instructable, part list Important: Localization: *Metric/Imperial units, Available materials and prices*
	CRAFT STUDENT	Nice designs and presentation for weekend project Good guarantee of successful builds: cost, instructable, part list *Important: Localization*
	PRO CRAFT	Complicated, profitable custom designs that are hard to design and prepared: staircases, cabinets, decks, sheds, houses Technical drawings, 3D models and calculations with sources for customization *Important: Localization*
	DESIGNER	Publish my cool designs for honor and profit Use parametric models as project templates
	DEV / DIGITAL MAKER	The most convenient way to 3D model with code and publish configurators Design automation toolset and platform (B2B) *Current core user!*
	BUSINESS	Offer customization options to customers in online configurators (existing industry: CPQ) production automation

1.2. USER ANALYSIS TAKE AWAYS

Based on our user interviews we generated these insights:

1. **Fragmented field of users**
   The mission of Archiyou needs to involve numerous users with different triggers, tooling and incentives

2. **Each user group needs something specific**
   Details are important to fit the needs of users. If you 3D print something is entirely different from DIY carpentry and needs different models and file formats. Also no imperial units can render documentation useless for US makers. *We now have a detailed view of what users need.*

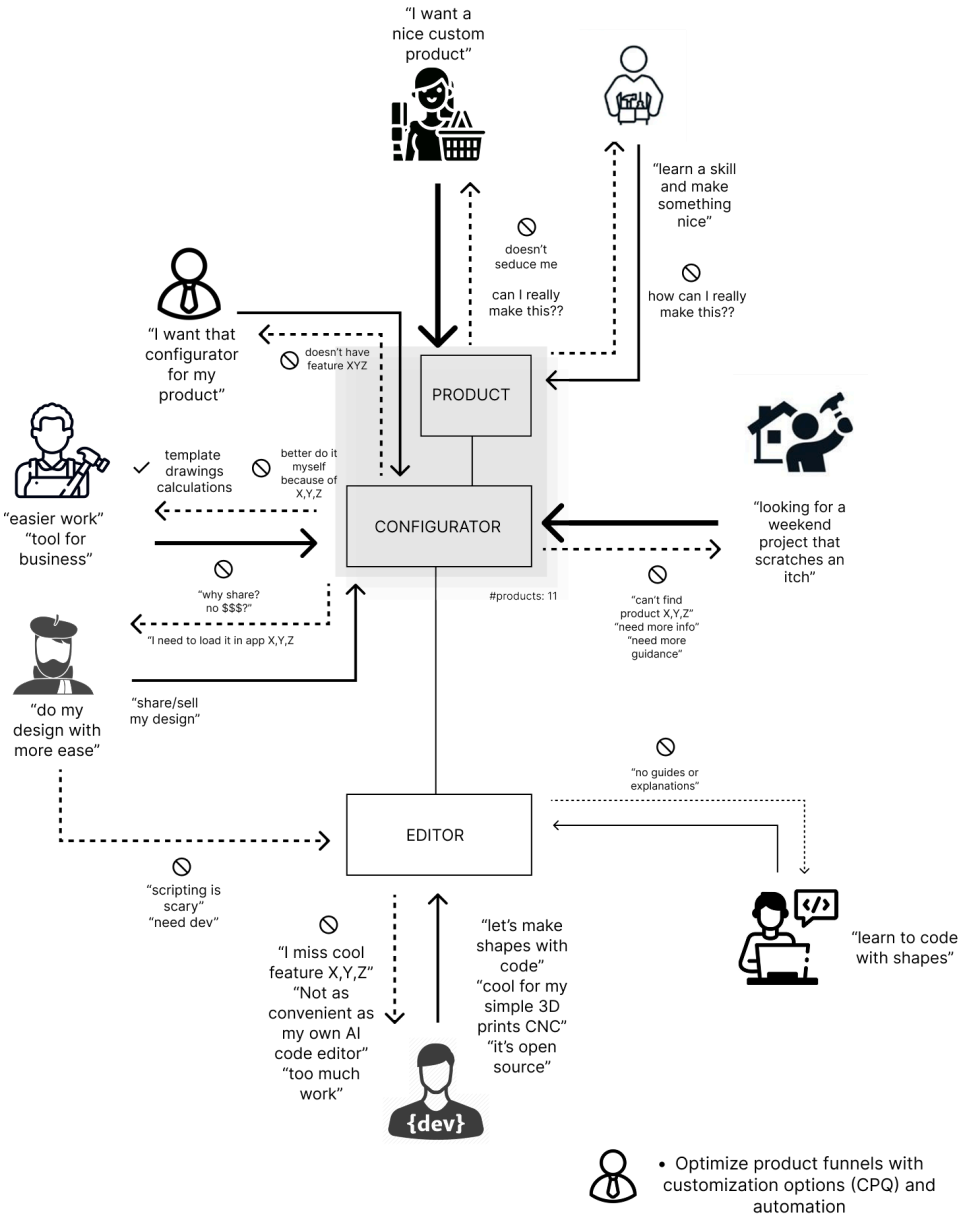
3. **Only early adopters are Developers / Digital Makers**
   Most user groups are not so tech-savvy and have specific needs (2) - We need to create the right fit for them if we want to serve a broader audience than the devs / digital makers.

4. **Relations between users by aspiration and expertise**

   There are interesting relationships between users. For example a consumer can turn into a DIY maker with the right documentation or get help from one. A craft student can grow into a PRO using Archiyou projects and can serve consumers more easily with our tools. Archiyou becomes more valuable if a developer can use it to automate for businesses.

5. **Minimal viable open design community**

   A lot of open source CAD software (from FreeCAD, OpenSCAD, CadQuery) and Archiyou too, serves a small community of early adopters but has difficulty serving a mainstream audience. We need to combine the various interests and expertises to trigger a flywheel to create a broader community.



User interviews: general triggers and blocks

1.3. COMPARITIVE ANALYSIS

In the last few years we have seen great (although mostly closed source) online design platforms gain traction to become (more or less) the de facto standard way of working.

We analyzed the following platforms for insights and lessons for our platform: Figma (2D UX/UI design and prototyping) , Penpot (open source 2D UX/UI design), Canva (Design everything) and Spline (3D design and publishing). We have also spoken to contributors of open source projects like FreeCAD/Lens for insights and analyzed 3D print content platforms like MakerWorld, Printables, Thangs and Thingiverse and others that offer DIY content like Instructables, Craftsy and other smaller ones.

UX Patterns & Consistency
Across successful platforms we identified common UX patterns that create clarity and consistency. Rather than reinventing the wheel, we adopted proven interaction models:

- Central canvas/viewer: Figma, Spline and Canva all place the design output front and center. Parameters and tools live in sidebars, never competing with the main visual.
- Real-time preview: Changes reflect immediately. Users never have to guess what their adjustments mean.
- Progressive disclosure: Canva excels at showing simple defaults while hiding advanced options. This serves beginners without limiting experts.
- Contextual help: Tooltips and inline guidance at the point of need, rather than separate documentation.
- Consistent component language: Buttons, inputs and controls behave predictably across all parts of the interface.

Community Features
We studied how platforms build and serve communities. Key patterns emerged:

- Showcase & discovery: MakerWorld, Printables and Thingiverse feature user creations prominently. This creates aspiration and triggers engagement.
- Remix culture: Platforms like Thingiverse allow users to build upon others' work. Penpot (open source) enables similar collaborative patterns.
- Social proof: Downloads, likes and "makes" (photos of built projects) create trust and validation.
- Creator profiles: Designers can build reputation and following, creating incentives for quality contributions.
- Collections & projects: Instructables and others let users group related items, enabling complete project workflows.

Feature Comparison Matrix

The following table maps common features across analyzed platforms and shows how they are implemented in Archiyou:

Feature	Figma	Canva	Spline	Maker World	Instr.	Archiyou Implementation
Central canvas/viewer	✓	✓	✓	✓	–	3D viewer dominates Configurator interface
Real-time preview	✓	✓	✓	✓	–	Parameters update 3D model instantly
Sidebar properties panel	✓	✓	✓	–	–	Left sidebar with grouped parameters
Slider + numeric input	✓	–	✓	✓	–	Dual input: sliders with direct value entry
Contextual tooltips/help	✓	✓	✓	✓	✓	Info tooltips per parameter with advice
Progressive disclosure	✓	✓	–	–	–	Basic vs. advanced parameters (collapsible)
Materials/parts list	–	–	–	✓	✓	Requirements section: wood, screws, tools
Cost estimation	–	–	–	✓	–	Real-time price based on parameters
Step-by-step instructions	–	–	–	–	✓	Instructable included in Pro Pack download
Multiple download formats	✓	✓	✓	✓	✓	DIY Drawing vs. Pro Pack with clear descriptions
User showcase/gallery	✓	✓	✓	✓	✓	Community section with user builds (planned)
Project collections	✓	✓	–	✓	✓	Project manager workspace (planned)
Localization (units/language)	✓	✓	–	✓	✓	Metric/Imperial toggle, language support

User Research Validation
The patterns identified in the comparative analysis were validated through user interviews. Key feedback that directly informed our design decisions:

User Feedback	Design Response
"I actually just want to see the product"	3D viewer takes center stage, text moved to sidebar
"It's unclear which properties mean what for the design"	Color coding links parameters to 3D model parts
"Sliders with an input field would be more user-friendly"	Dual input: sliders combined with numeric entry
"Everything is now a mix of English and Dutch"	Consistent localization throughout interface
"I can't do anything with the manual... which screws, the steps"	Complete requirements section with materials, fasteners, tools
"What's the difference between DIY and Pro Pack?"	Clear download options with explicit content descriptions
"I only expect length, width, height"	Basic parameters visible, advanced options collapsible

Conclusions
The comparative analysis of both commercial and open source platforms has yielded valuable insights that were directly applied in the Archiyou Configurator redesign. By combining established UX patterns with specific user research feedback, we created an interface that:

1. **Feels familiar** to users coming from other design tools
2. **Serves diverse user groups** from consumers to professionals
3. **Builds trust** through transparent information on costs and requirements
4. **Enables community growth** through showcase and sharing features

This design supports Archiyou's strategic shift from a developer-focused platform to one that can serve a broader open design community.

2. STRATEGY AND ROADMAP FOR 2026

Based on the user interviews, the comparative studies and team breakout sessions we formulated the following strategy:

1. To each his own

A great understanding of users helps it to serve them better in the coming Developments. Greatly simplified we formulated these emphasis per user group:

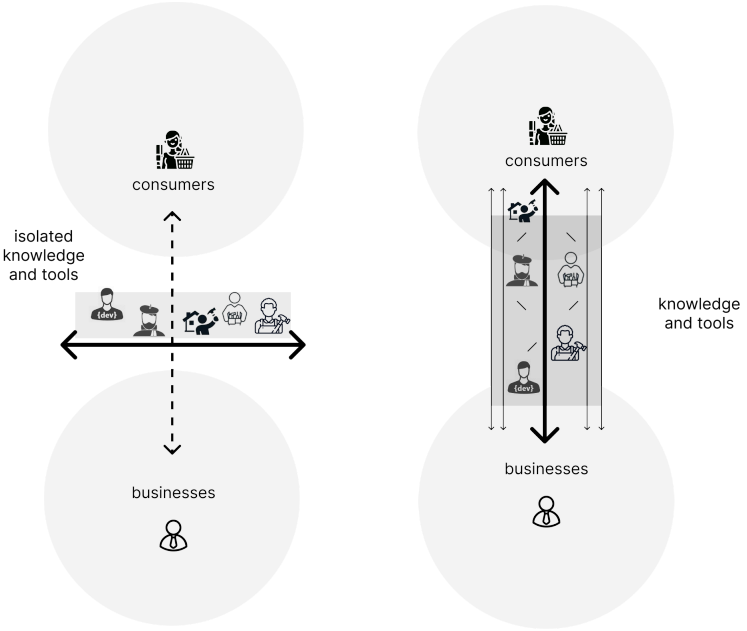
Consumer	*Great custom designs presented in a non-technical way and ways to get them (DIY, kit, pro-made)*
DIY maker	*Great weekend projects and their step-by-step instructable*
Craft student	*Make a nice product and learn a skill at the same time*
Pro Craftsman	*Make designing, selling and building and complex projects easy. All files you need to make it your own.*
Designer	*Use parametric designs as a basis for your own. Make your design process easier. Publish your design for glory and profit.*
Developer / Digital Maker	*Open source design modeling, automation and configurators*
Business	*Open source, portable and customizable automation and configurators (CPQ)*

2. The extra mile to the user

If it's easy just implement as many need-to-haves for every user group
Short term: Very easy hotfixes already resulted in big usage increases
Long term: Keep a realistic priority what users to be able to serve

3. Connect early adopters to the real world (of consumers and producers)

Connect small isolated group of early-adopters to the "real world" of consumers an producers. This creates triggers, incentives and aspirations.
Archiyou already offers developers / digital makers a fully online ecosystem to take a parametric design to the web. But we need to achieve a more robust open source basis and extend functionality.



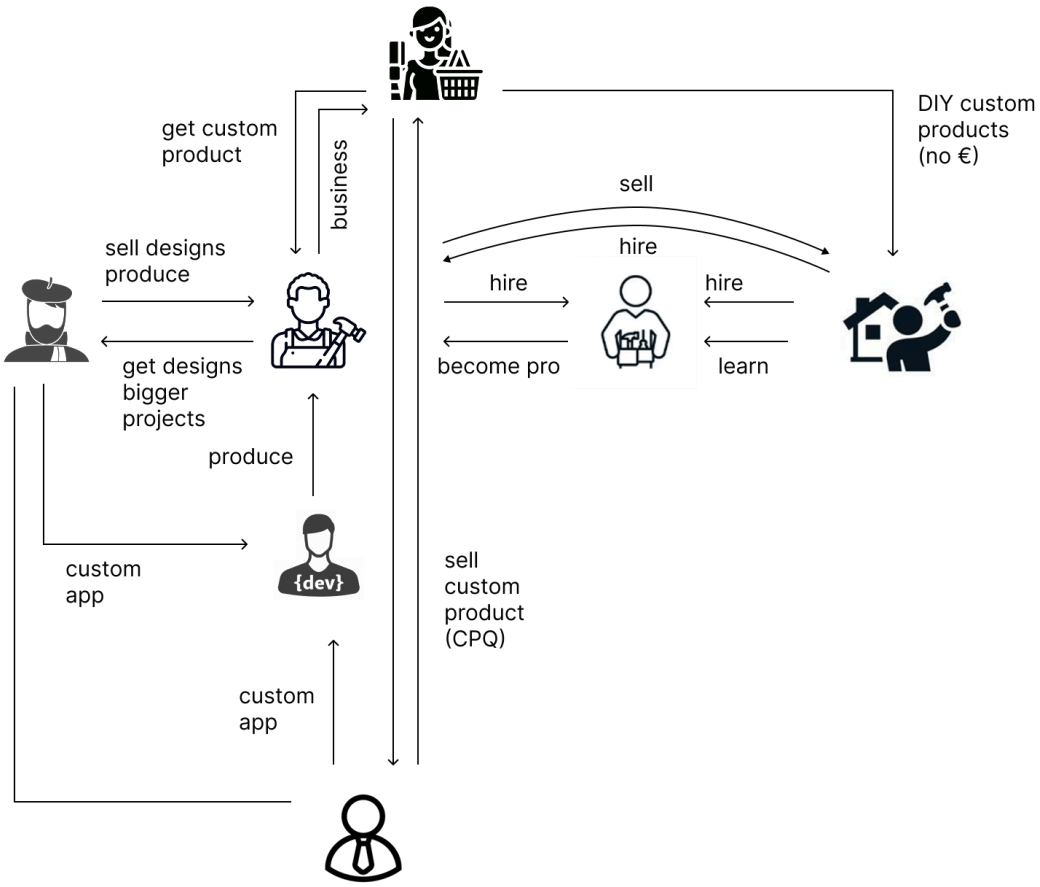
4. Doing projects on Archiyou

While one part of Archiyou is very much about creating parametric models in the Editor. The other part is all about content: Customizing those designs in the configurators and applying them to a project: for yourself or clients.

We will introduce a workspace and project management layer that gives designers, DIY makers, student and craft professionals the tools to manage their projects, the documentation outputs and interactions with clients. In this way we can broaden our user base.

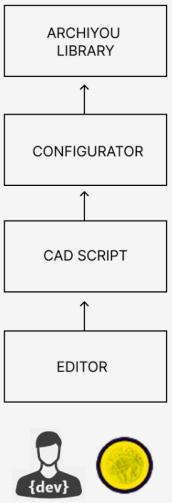
5. From users to community

Besides serving the individual needs of users, we will prioritize features that also support mutually beneficial relationships. This is a map of those possible relationships:

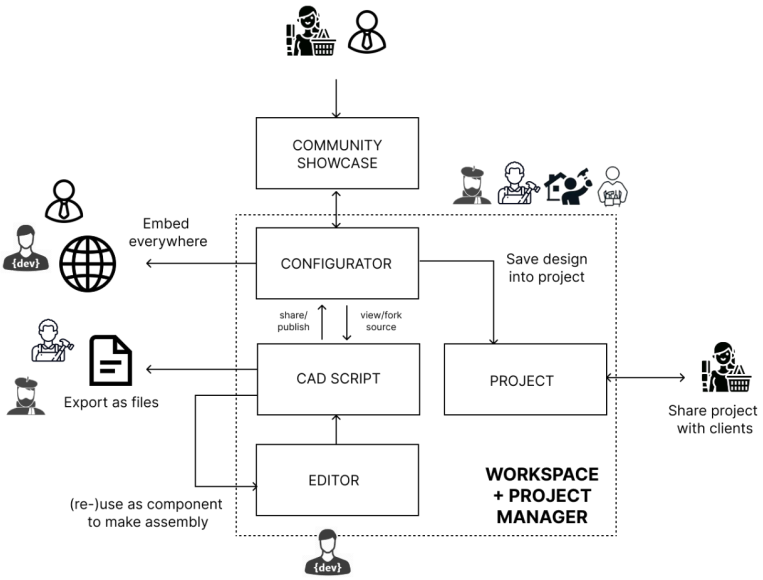


6. From open core tooling to open source design

In a lot of ways Archiyou functions as an open-core platform, where its technology is mostly used by us and for us. The previous analysis and strategic approaches result in a vision for an open source design community:



FROM OPEN-CORE
INTERNAL TOOLING



TO OPEN SOURCE/DESIGN
COMMUNITY

3. UX/UI DESIGN

The UX/UI design phase translated the insights from our user research, comparative analysis and strategic roadmap into concrete, high-fidelity interface designs and covers the three main touchpoints of the Archiyou platform: the Configurator (public-facing parametric design tool), the Editor (script authoring environment) and the Workspace (project and asset management application).

Each design was developed iteratively through user feedback sessions, breakout workshops with developers and designers, and continuous validation against the user groups identified in Section 1. The designs are delivered as high-fidelity Figma files with full interaction specifications, accompanied by detailed experience rationale documents that explain the reasoning behind every design decision.

The designs shown in this document are simplified. Please see https://github.com/ArchiyouApp/archiyou-core/documentation/nlnet/uxui for high-fidelity designs and reasoning.

3.1. DESIGN PRINCIPLES

From our user research and comparative analysis we distilled a set of core design principles that guided all interface decisions across the platform. These principles directly address the challenges identified in Section 1: serving a fragmented user base, bridging the gap between developers and non-technical users, and supporting the transition from open-core tooling to an open design community.

The core design principles act as a system of beliefs and define how design should function across the Archiyou platform. They are applied across the Configurator, Editor, and Workspace and have directly informed every decision presented in the sections that follow.

1. Progressive Disclosure
Show simple defaults first; reveal complexity on demand. This principle directly responds to user feedback that interfaces felt overwhelming. Validated by our comparative analysis of Canva and Figma, which excel at hiding advanced options while keeping them accessible. In the Configurator this manifests as the Basics/Materials/Advanced tab structure; in the Editor as collapsible panels for code, console and data.

2. Real-time Feedback
Every parameter change reflects immediately in the 3D viewer. Users never have to guess what their adjustments mean. This was the single most validated pattern across all platforms we studied (Figma, Spline, Canva) and was explicitly requested in user interviews: “Sliders with an input field would be more user-friendly.”

3. Guidance Over Restriction
Provide contextual help at decision-heavy moments without enforcing defaults. Rather than hiding complexity or making choices for users, we help them make informed decisions. This manifests in

features like “Help me choose standard sizes” links, contextual AI assistance in the Editor, and standard presets that serve as starting points rather than constraints.

4. Separation of Concerns
Clearly separate tools (scripts), results (designs and outputs), and project context across the interface. This reduces cognitive load and prevents the confusion between source and output that users reported. In the Workspace, this is reflected in distinct card types; in the Editor, through the clear division between parameters, code and geometry.

5. Intent-Driven Organization
Organize content and actions by what users want to achieve, not by technical structure. Export options are grouped as fulfillment paths (“Make it yourself”, “Produce with a machine”, “Design & modify”) rather than file formats. The Workspace organizes by intent (design files, scripts, projects) rather than file type.

6. Open by Default
Make openness and reuse explicit throughout the interface. “View source”, “Embed”, and “Fork to edit” actions are always visible, supporting open-source and remix workflows. Shared scripts show clear author attribution. Publishing is an explicit, controlled decision that supports gradual transition from private exploration to community contribution.

3.2. CONFIGURATOR / VIEWER REDESIGN

The Configurator is the public-facing interface where end-users interact with parametric designs. It is the primary touchpoint for our broadest user groups: consumers, DIY makers, craft students and professionals. The redesign was driven by the core finding from user research that “most user groups are not so tech-savvy and have specific needs” and that “details are important to fit the needs of users.”

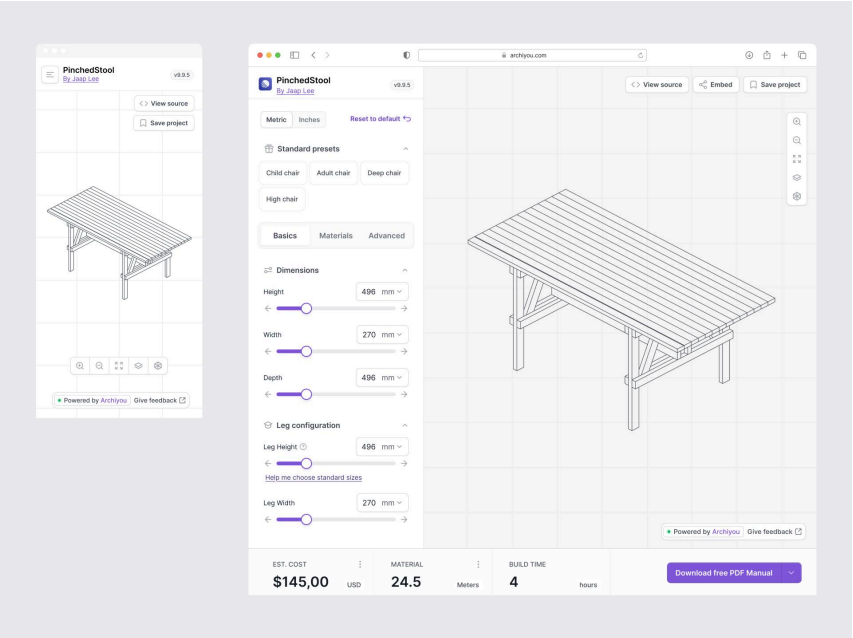


Figure 3.1 — Configurator redesign: desktop layout with parameter sidebar, live 3D preview and action bar

Layout and Interaction Model

The Configurator follows the established pattern of a central canvas/viewer with a sidebar for parameters, directly informed by our comparative analysis of Figma, Spline and Canva. The 3D viewer takes center stage as the primary element, responding to user feedback: “I actually just want to see the product.” Parameters live in a left sidebar that never competes with the 3D preview. The bottom action bar provides derived metrics (estimated cost, material usage, build time) and the primary download action, completing the design-to-build workflow.

Parameter Design

Dimensional parameters combine sliders for rapid exploration with numeric inputs for precise specification, directly implementing user feedback about wanting “sliders with an input field.” Each parameter includes a unit selector (mm by default) with Metric/Inches toggle at the top level, addressing the critical localization need identified across all non-developer user groups. Parameters are grouped by physical concepts (Dimensions, Leg configuration) matching how users think about objects being built, rather than by technical parameter names. Standard presets (e.g. Child chair, Adult chair) function as onboarding shortcuts: after selecting a preset, all parameters remain fully adjustable, reinforcing that presets are starting points rather than constraints.

Progressive Disclosure Through Tabs

The Basics/Materials/Advanced tab structure manages complexity for our diverse user base. The Basics tab shows fundamental dimensions that users reported expecting (“I only expect length, width, height”). The Materials tab exposes craft-specific parameters like wood type, hardwood/softwood selection and exterior use properties. The Advanced tab contains specialized configuration including object-level customization with file uploads, text properties and nested parameters. This three-tier structure ensures that consumers and DIY makers see an approachable interface, while designers and professionals can access full parametric control.

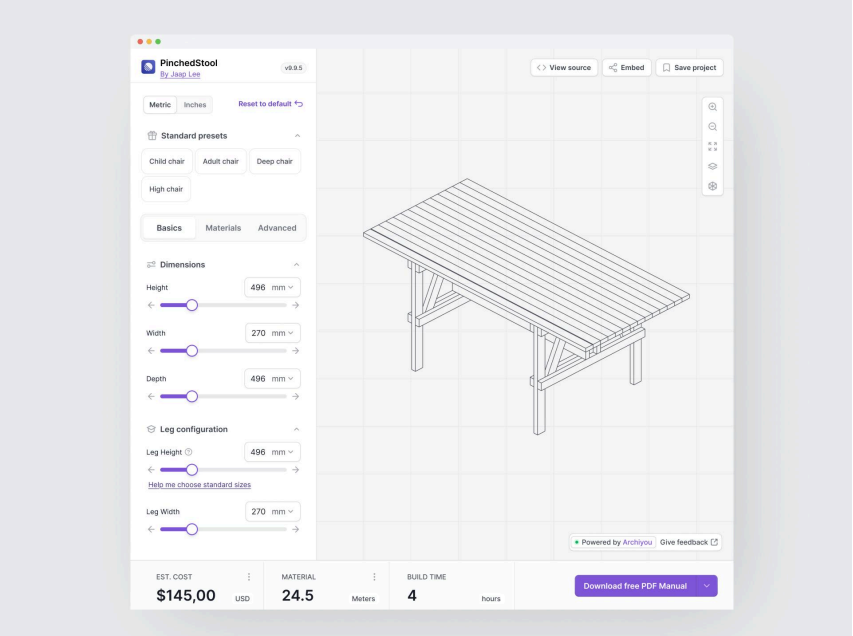


Figure 3.2 — Configurator responsive design: embeddable widget (left), browser view with parameters (center), full desktop view with metrics bar (right)

Fulfillment Layer

A key innovation in the redesign is the fulfillment layer, which translates configured designs into concrete outputs. Export options are organized as fulfillment paths rather than file formats, framing outputs in terms of user intent: “Make it yourself” (DIY instruction manual, technical drawing, complete DIY pack), “Produce with a machine” (3D print STL, 2D/3D CNC milling files), “Design & modify” (CAD formats for Fusion/SolidWorks/FreeCAD, Sketchup, BIM models), and “Visualize & present”. This directly addresses the user research finding that different user groups need entirely different output formats: someone who 3D prints needs something entirely different from DIY carpentry.

Derived Metrics

Estimated cost, material usage and build time are presented as derived outcomes of design decisions in the bottom action bar. These metrics update in real-time as parameters change, making trade-offs visible and supporting informed decision-making. This feature directly serves DIY makers and craft professionals who need cost and material estimates before committing to a build, addressing the user need for “good guarantee of successful builds: cost, instructable, part list.”

Embeddable and Portable Design

The Configurator is designed in three responsive tiers: a compact embeddable widget for third-party websites, a standard browser view, and a full desktop view. The embeddable variant features a “Powered by Archiyou” attribution and minimal controls (View source, Save project), supporting the strategy of connecting early adopters to the real world by allowing developers to embed configurators in their own websites. The “View source”, “Embed” and “Save project” actions make openness and reuse explicit throughout all variants.

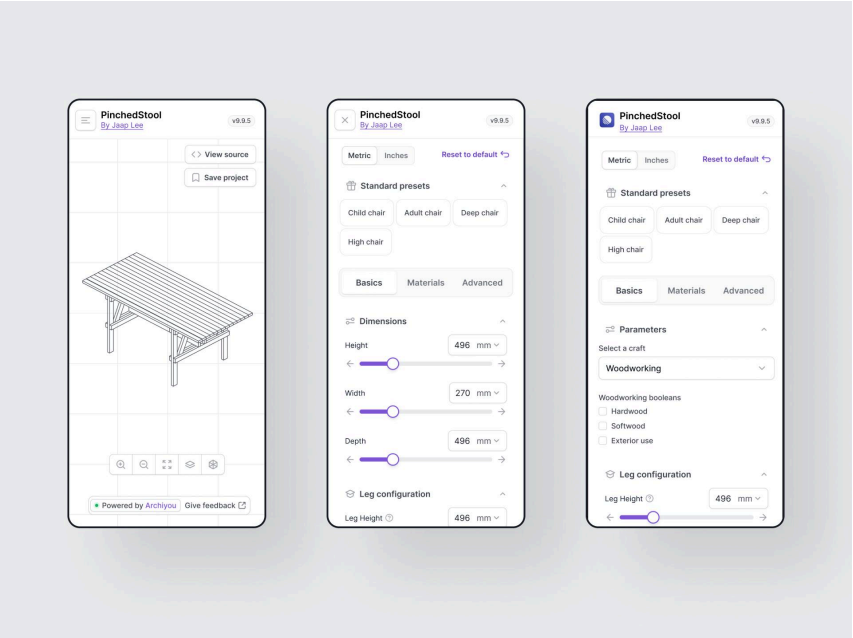


Figure 3.3 — Configurator mobile design: 3D viewer priority (left), parameter panel with dimensions (center), advanced parameters (right)

Mobile Design
The mobile Configurator was designed to serve the large audience of consumers and DIY makers who primarily browse on phones. On mobile, the 3D viewer is given full-screen priority, allowing users to immediately see and manipulate the object before accessing parameters. Configuration options are presented in a dedicated scrollable panel that slides over the viewer when activated. The Basics/Materials/Advanced tab structure is preserved to maintain consistency across devices, reducing relearning and supporting cross-device workflows. All interactive elements (sliders, inputs, buttons) are sized and spaced for reliable touch interaction without sacrificing precision.

3.3. EDITOR REDESIGN

The Editor is where developers and designers create parametric scripts. It is the creative engine of the platform, serving the Developer/Digital Maker user group (current core users) while also needing to become more accessible to designers who want to publish and share their work. The redesign focuses on making the Editor a unified environment where parameters, code and geometry coexist, supporting both exploratory design and learning-by-doing.

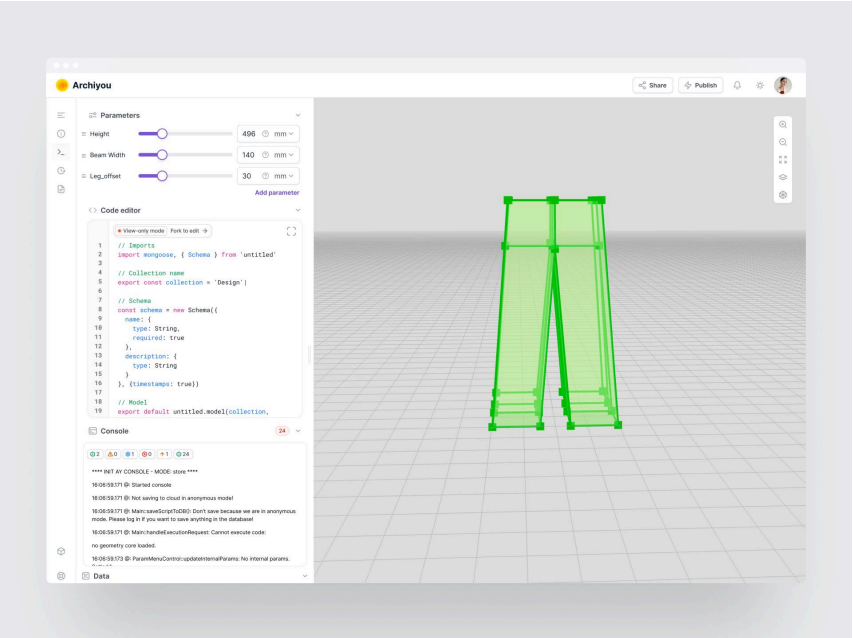


Figure 3.4 — Editor redesign: unified environment with parameters, code editor, console, and live 3D preview

Interaction Model

The Editor separates control (parameters), logic (code) and result (geometry) to reduce cognitive load while supporting complex workflows. The left panel contains the parameter controls and code editor in a vertically stacked, collapsible layout. The right panel is dominated by the live 3D viewer that reflects parameter and code changes instantly. This layout reinforces the cause-and-effect relationship between inputs and outputs, which is fundamental to parametric design thinking. The console provides real-time feedback and errors below the code editor, making system state visible during iteration.

Parameter System

Parameters provide a structured control layer that exposes the key variables driving the design. Each parameter combines a slider for exploration with a numeric input for precision, consistent with the Configurator design language. An “Add parameter” action allows the design system to grow dynamically. Parameters include info tooltips and unit selectors, maintaining the guidance-over-restriction principle. This system bridges the gap between the code (where parameters are defined) and the visual result (where parameters have effect), making scripts accessible to non-coders who receive shared scripts.

Code Editor

The code editor exposes the underlying logic of the design with syntax highlighting, line numbers and intellisense support. A key design decision is the “View-only mode / Fork to edit” toggle, which enables safe inspection of shared scripts while supporting remix workflows through forking. This directly supports the open design community strategy: users can browse, learn from and build upon others’ work without affecting the original.

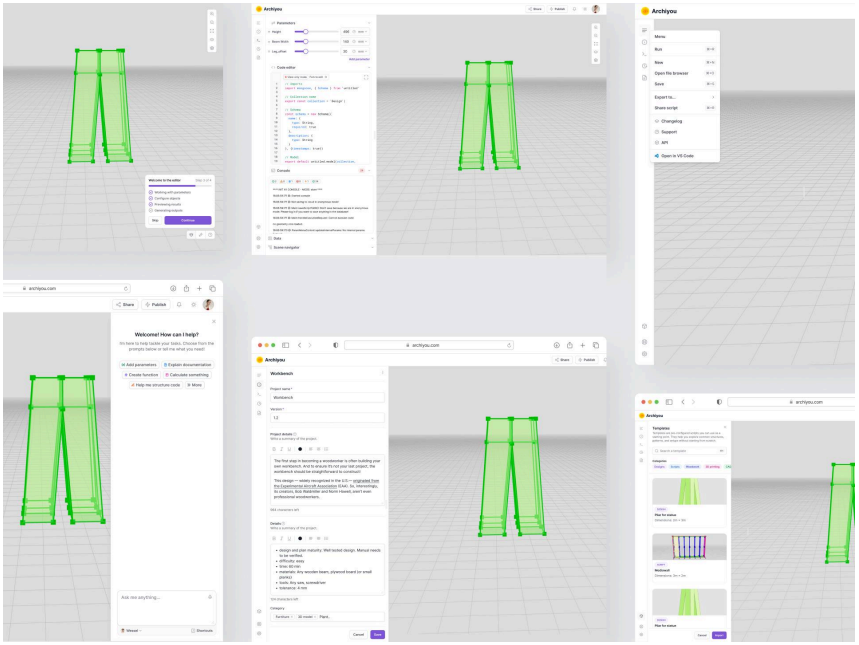


Figure 3.5 — Editor feature set: menu system, onboarding flow, AI assistance, template library, publishing and sharing dialogs

Onboarding
The onboarding system guides new users through the Editor by introducing core concepts (parameters, objects, outputs) step by step as a lightweight overlay, allowing users to continue exploring while being guided. The onboarding is divided into clear steps reflecting the core workflow from configuration to output. Users can skip or resume based on their experience level. This approach supports learning-by-doing without forcing a linear tutorial, which was identified as critical for serving the range from beginners to experienced developers.

Contextual AI Assistance (not part of NLNET project)
The AI panel provides task-oriented assistance directly within the Editor. Suggested actions (“Add parameters”, “Explain documentation”, “Create function”, “Calculate something”, “Help me structure code”) focus on common design tasks, lowering the barrier to asking the right question. Rather than hiding complexity, the AI explains decisions and structures, reinforcing understanding. The panel is always available but never blocks the primary workflow. This feature helps bridge the gap between the technical script environment and users who are more design-oriented.

Examples/Template Library
The template library provides curated starting points organized by category (Designs, Scripts, Woodwork, 3D printing, CAD). Templates are presented as editable starting points rather than finished products, encouraging exploration and modification. Each template shows a visual preview combined with key dimensions to set expectations early. By starting from an existing structure, users focus on understanding and modification rather than setup, which aligns with our strategy of lowering the barrier to entry for non-developer user groups.

Sharing and Publishing
The design introduces a clear two-stage model for making scripts available to others. Sharing is framed as an extension of learning, using email-based invites to collaborate on specific scripts. Publishing is a separate, explicit decision with clear toggles for controlling how others can interact with the script (view, modify, share). Making a script public enables discovery within the community

without requiring direct sharing. This graduated approach supports the strategy of building a community where users can move from private exploration to public contribution at their own pace.

Project Identity and Publishing Flow
The publishing dialog captures essential project metadata: name, version, description, difficulty, time estimate, materials and tools needed, and categories. This structured information directly supports the needs of downstream users identified in our research: consumers need to know difficulty and materials before deciding to build, DIY makers need step-by-step details, and professionals need technical specifications. Categories (Furniture, 3D model, Plant, etc.) improve discoverability within the community library.

3.4. WORKSPACE / APPLICATION REDESIGN

The Workspace is the project management layer that connects all parts of Archiyou into a unified application. It is a new addition to the platform, directly implementing Strategy point 4 (“Doing projects on Archiyou”) from Section 2. The Workspace gives designers, DIY makers, students and craft professionals the tools to manage their projects, documentation outputs and interactions with collaborators.

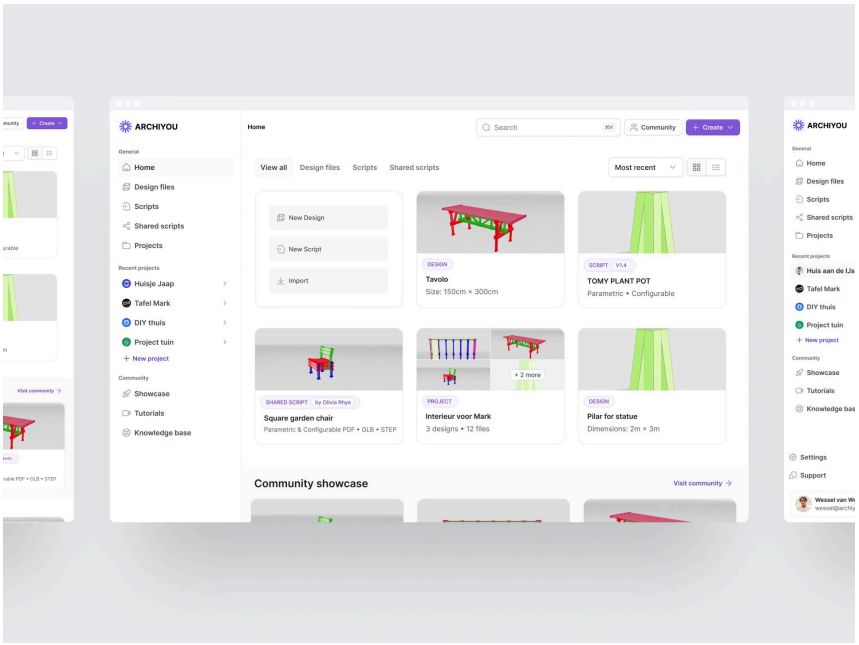


Figure 3.6 — Workspace home screen: unified overview of designs, scripts, shared resources and community showcase

Home and Navigation
The Home screen functions as a workspace overview, giving users immediate access to active projects, reusable scripts, design files and shared community resources. The left sidebar provides persistent navigation organized into three zones: General (Home, Design files, Scripts, Shared scripts, Projects), Recent projects (with color-coded project indicators), and Community (Showcase, Tutorials, Knowledge base). Content is organized by intent rather than technical structure: producing design files, creating scripts, reusing shared scripts, or managing project context. A Community

showcase section is visually separated from personal workspace content, positioning it as an inspiration and discovery layer.

Content Type System
Each card in the workspace explicitly communicates its object type through labeled badges: Design (tangible results with dimensions), Script (versioned, configurable building blocks), Shared Script (community contributions with author attribution), and Project (aggregations of multiple designs and files). This type system enables recognition over recall, prevents conceptual overlap between source and output, and makes the distinction between tools and results immediately clear. Design files emphasize tangible properties like dimensions, scripts highlight versioning and configurability, and projects show scope indicators (“3 designs, 12 files”).

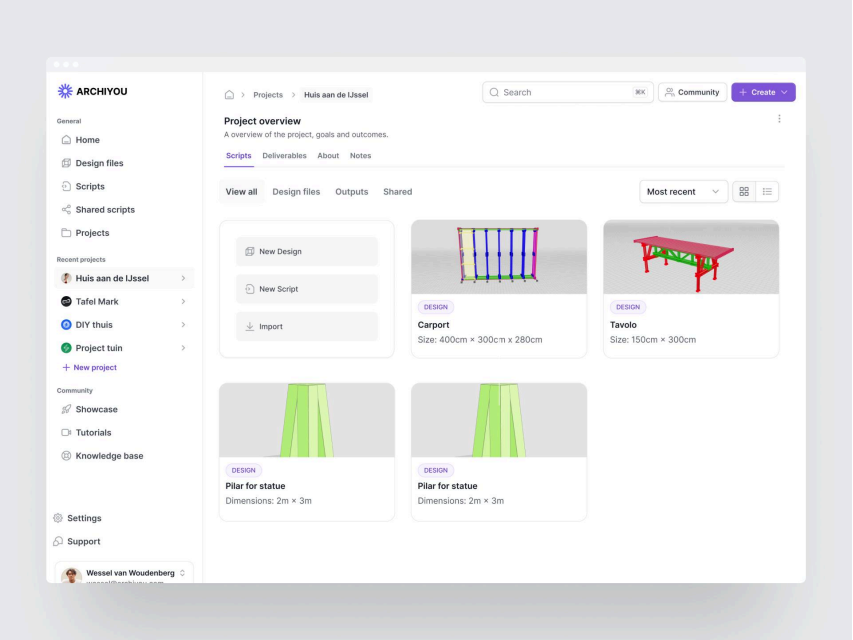


Figure 3.7 — Project overview: scripts, deliverables and design files organized within a project context

Project Overview
The project overview frames designs, scripts and outputs within a shared project context, positioning the project as the primary unit of work rather than individual files. Projects are organized through tabs (Scripts, Deliverables, About, Notes) and content filters (View all, Design files, Outputs, Shared). The creation card offers three clear entry points: New Design, New Script, and Import. Individual designs are presented as discrete, inspectable artifacts with visual previews and metadata (type badge, name, dimensions), supporting fast scanning and recognition. This structure supports long-running, iterative projects and reflects real-world design and construction workflows where multiple related designs coexist.

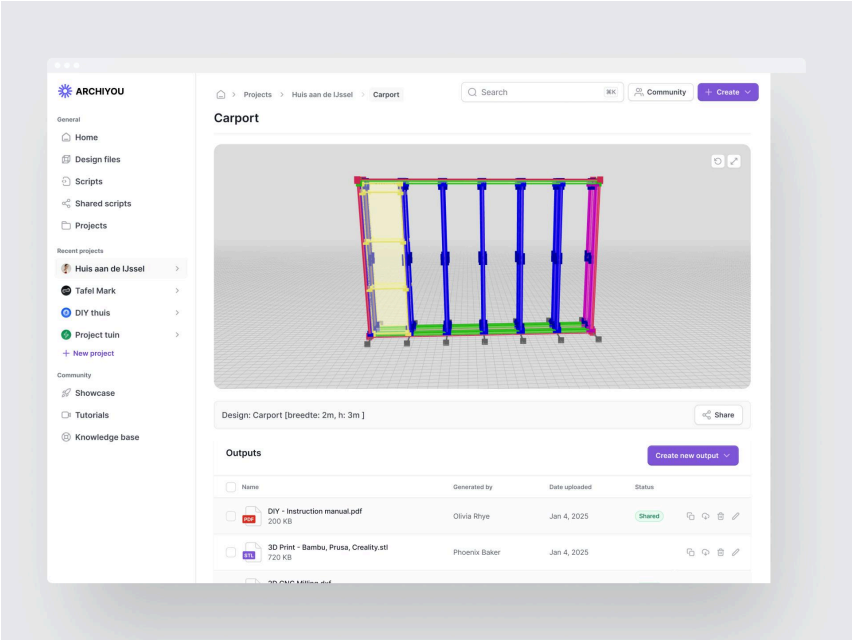


Figure 3.8 — Script detail page: 3D preview, output management with categorized export options

Script Detail and Output Management

The script detail page positions scripts as reusable capabilities that generate designs and outputs. The central 3D preview provides immediate visual context. Below the preview, a comprehensive output management system allows users to generate, track and share deliverables. The output options panel organizes exports by fulfillment intent (Make it yourself, Produce with a machine, Design & modify, Visualize & present), consistent with the Configurator design. Generated outputs display ownership, sharing status and timestamps, making collaboration explicit. Breadcrumbs and persistent navigation make it clear how the script fits within the broader project structure.

3.5. EXTRA: LANDING PAGE AND PUBLIC COMMUNICATION

The landing page design translates the platform’s value proposition into a clear, non-technical narrative that speaks to the broadest possible audience, particularly the consumer and DIY maker groups.

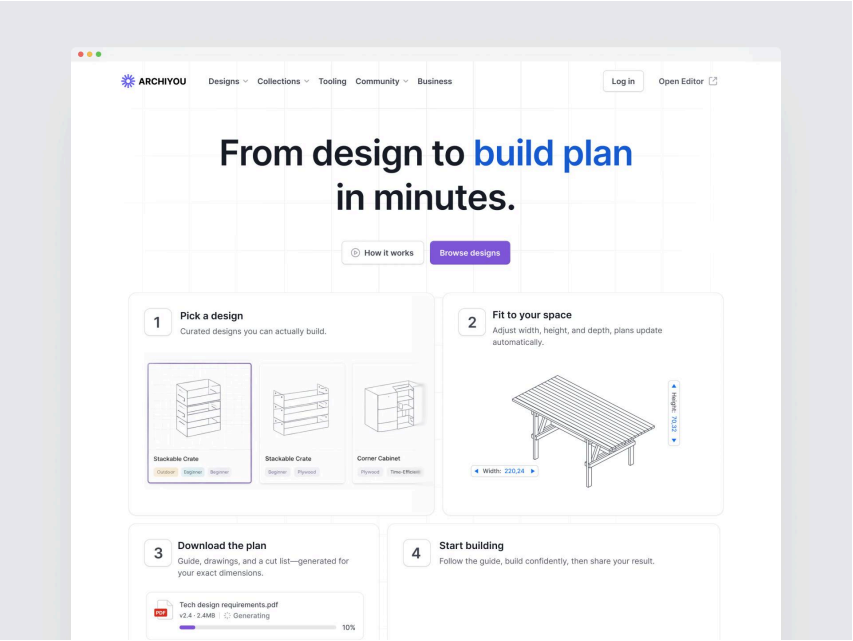


Figure 3.9 — Landing page: four-step value proposition communicating the design-to-build workflow

The headline “From design to build plan in minutes” captures the core value proposition in non-technical language. The page communicates the platform through a four-step narrative: (1) Pick a design from curated, buildable options, (2) Fit to your space by adjusting parameters with live preview, (3) Download the plan with generated guides, drawings and cut lists, and (4) Start building with confidence. This narrative directly addresses the consumer insight that they want a “good guarantee of successful builds” and reframes parametric design from a technical concept into an accessible tool. The navigation (Designs, Collections, Tooling, Community, Business) reflects the intent-driven organization principle, providing clear entry points for different user groups.

3.6. DESIGN SYSTEM AND VISUAL LANGUAGE

Across all redesigned interfaces, a consistent visual language ensures coherence and reduces cognitive friction when users move between the Configurator, Editor and Workspace.

Visual Identity The design system uses a clean, neutral interface palette with a single accent color applied consistently for interactive elements (sliders, active tabs, primary buttons, links). In the current high-fidelity mockups, purple is used as a working accent color to clearly distinguish interactive elements during the design phase. This is not the final brand color, the production implementation will adopt the established Archiyou brand identity and color palette. The design system is built to be color-agnostic: swapping the accent color requires no structural changes, as all interactive elements reference a single design token. The Archiyou wordmark and icon appear consistently across all interfaces. Typography follows a clear hierarchy with consistent sizing across components. The overall visual tone is professional but approachable, avoiding the overly technical aesthetic common in CAD tools.

Component Consistency Key interaction patterns are consistent across all interfaces: sliders with numeric inputs always behave the same way (Configurator, Editor); card components follow the same structure (preview, type badge, title, metadata) in both Workspace and Community; collapsible sections use the same chevron pattern throughout; and action buttons follow a

consistent hierarchy (primary filled, secondary outline, tertiary text). This consistency was directly informed by the comparative analysis finding that successful platforms like Figma use "consistent component language where buttons, inputs and controls behave predictably across all parts of the interface."

Responsive Design Strategy All interfaces are designed for three breakpoints: mobile (Configurator-focused), tablet and desktop. The sidebar navigation in the Workspace collapses to icons on smaller screens while maintaining the same information hierarchy. The Configurator adapts from a side-by-side layout on desktop to a stacked layout on mobile with the 3D viewer taking priority. This ensures that the growing mobile audience identified in our analytics can access the full platform experience.

4. TECHNICAL SPECIFICATION

Based on our strategy we distinguish the following various key components within the Archiyou platform and identify the most important technological requirements and choices:

#	part	subpart	description	reqs & specs	choices
1	Archiyou core		Runs Archiyou scripts in a variety of contexts. Can be used as TS/JS/WASM module from NPM	Portable NPM module	Open source as module Improve devx
		Existing BREP Geometry kernel	Generates all simple and complex geometry now based on OpenCascade/WASM. Slow and brittle.		Lessen dependency with new kernel Implement switching logic
		New MESH Geometry kernel	High performance simple Mesh geometry	High performance General purpose Document-specific functions: projections, exports	CSGRS/WASM New: Meshup TS/JS library
2	Archiyou backend		Backend of current editor ecosystem and publishing infrastructure		
a		Archiyou publish	Backend serving and executing of Archiyou scripts REST API	Good integration with archiyou-core Horizontal scaling with Workers Task queuing	Nodejs Fastify Redis/BullMQ
b		Archiyou workspace server	Backend of editor ecosystem. Saves script versions, shares scripts etc. REST API. New: project and community functions (see 3) *Currently: Postgres DB, Python FastAPI*		NodeJS/Fastify like publish, with possible extra file-based storage system Will be combined with Archiyou publish for ease of use and maintainability
3	Configurator / Viewer		Visual side of Archiyou script: Present a parametric model. User can change it by modifying parameters. Archiyou script is run by an Archiyou Publish instance (2a.). User can request a variety of exports: Like PDF plans. *Currently an old heavy Nuxt2/Vue2 app*	Portable Embeddable Light-weight JS framework independent (use in React/Vue/Svelte etc) Decent 3D viewing with extra features like AR	Webcomponents Lit framework Shoelace/WebAwesome UI components ThreeJS
4	Archiyou workspace (client side)				
a		Editor	Users create the Archiyou scripts, debug and view results *Currently: slow and bulky Monaco/VSCode with*	Light-weight speed Visual feedback and integrated help	CodeMirror Webcomponents / Lit Shoelace/WebAwesome

			Nuxt2/Vue2 and old Bulma/BuefyUI components	Extendable (add extra code tools)	
		Community	New sharing and showcase functionality		Webcomponents / Lit Shoelace/WebAwesome
		Project manager	New way to manage projects from configurator, asset management and sharing project/designs/outputs		

All components will be made open source (Apache2). Starting with the core - which already is.