

Landing Page Generator

Product concept, interface direction & implementation blueprint



A lightweight PHP mini-SaaS that helps creators, marketers, and agencies build, customize, preview, and export polished landing pages—without depending on a hosted page-building service.

5	2	ZIP	PHP + MySQL
Design templates	Languages	Portable export	Core stack

Visual direction

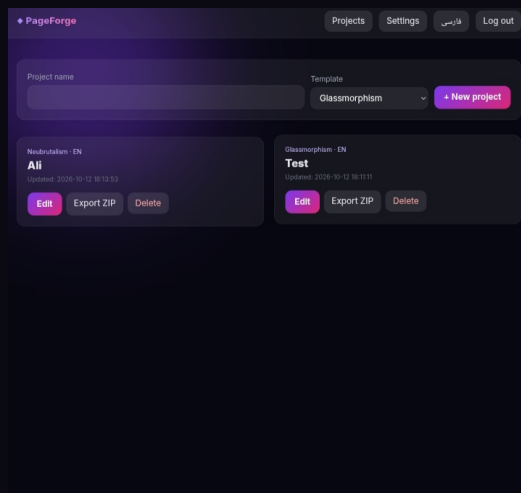
Dark interface • glass-like panels • purple-to-magenta accents • rounded controls • compact typography • responsive layouts

Prepared from the supplied PageForge dashboard, settings, and editor screenshots, together with the product requirements in the brief.

01 / Dashboard & account settings

The dashboard should make project creation and ongoing management immediately obvious. Account settings stay lightweight and accessible from the top navigation.

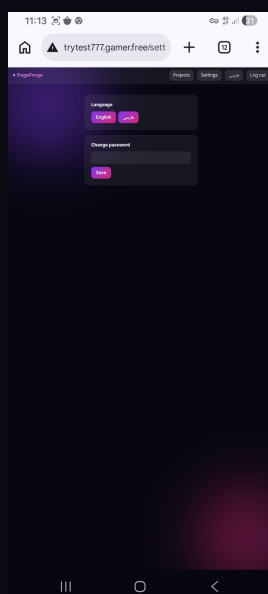
Dashboard reference



Core dashboard behavior

- **Project creation:** enter a project name, choose a template, then create the project.
- **Project cards:** show template, language, last-updated time, and clear Edit, Export ZIP, and Delete actions.
- **Account navigation:** keep Projects, Settings, language switching, and Log out visible in the header.
- **Responsive behavior:** stack cards on narrow screens, maintain touch-friendly buttons, and prevent horizontal overflow.

Settings reference



Settings should support English/Persian selection and password changes. Password updates must verify the current session, validate the new password, and use PHP password hashing.

02 / Editor experience

The editor is the main workspace. Keep the form organized into short sections, use repeatable item cards for content lists, and provide persistent Save and Export actions.

PageForge Projects Settings فارسی Log out

Save Export ZIP

Template: Glassmorphism Page language: English

Brand name: Brand

Logo: Choose file No file chosen

Background image: Choose file No file chosen

Hero title: Launch your idea this week

Hero subtitle: A fast, beautiful page that turns visitors into customers.

Button text: Get started

Button link: #pricing

Footer: © 2026 Brand

Features: + Add

- Fast setup: Everything you need, nothing you do not.
- Fully responsive: Everything you need, nothing you do not.
- Easy to edit: Everything you need, nothing you do not.

Required editor sections

- **Project setup:** template and page language.
- **Brand:** brand name, logo upload, background image upload.
- **Hero:** title, subtitle, CTA label, CTA URL.
- **Features:** icon, title, description; add/remove items.
- **Testimonials:** person, role, quote; add/remove items.
- **Pricing:** plan name, price, feature list; add/remove plans.
- **Footer:** customizable footer text.
- **Preview:** reflect edits immediately without losing form state.

Usability notes: label every input, preserve entered data after validation errors, show upload limits beside file controls, and confirm destructive actions such as deleting a project.

03 / Templates, language & RTL

One shared renderer should power five distinct visual themes. Each theme can vary typography, spacing, surfaces, borders, gradients, and motion while using the same underlying project configuration.

TEMPLATE	VISUAL CHARACTER	DESIGN NOTES
Glassmorphism	Translucent surfaces, blur, soft borders	Primary build target; elegant cards and gradient glow.
Aurora UI	Fluid luminous color fields	Layered gradients and subtle ambient lighting.
Neubrutalism	Bold outlines, strong contrast	Confident typography, hard shadows, playful blocks.
Minimalist Dark	Quiet, precise, high-contrast	Reduced decoration and clear typographic hierarchy.
3D Gradient	Dimensional color and depth	Layered gradients and polished depth cues without heavy libraries.

English + Persian requirements

- Translate every dashboard label, validation message, button, and status string through a simple PHP localization array in **lang.php**.
- Store the selected page language in the project JSON configuration so the exported site preserves its language.
- For Persian, set **lang="fa"** and **dir="rtl"**; mirror layout alignment and spacing where appropriate.
- Use Vazirmatn for Persian typography, loaded from a CDN in the generated page. Keep a sensible fallback font stack.
- Ensure mixed-direction content such as URLs, prices, and email addresses remains readable in RTL layouts.

The dashboard language preference and the exported landing-page language are related but separate settings: users may manage the app in one language while configuring a page in another.

04 / Architecture & delivery plan

Build for ordinary shared hosting: no heavy framework, no required Node build step, and no external cloud dependency. Tailwind CSS may be used via CDN in the dashboard; exported pages should include their own CSS and JavaScript assets.

Suggested file map

config.php / lib.php	Database credentials, PDO connection, escaping, CSRF and shared helpers.
auth.php	Registration, login, logout, session handling, password hashing.
dashboard.php / editor.php / settings.php	Project management, editor form, and user settings.
preview.php / render.php	Preview endpoint and shared landing-page renderer.
export.php / upload.php	ZIP export and safe image upload processing.
lang.php / setup.php / database.sql	Localization, optional table setup, and import-ready schema.
uploads/ / README.md	Protected upload directory and installation guide.

Security & portability checklist

- Use PDO prepared statements, password_hash()/password_verify(), session hardening, and CSRF tokens for state-changing requests.
- Every project query must be scoped to the authenticated user ID; never trust a submitted project ID alone.
- Validate upload MIME type and extension, cap file size at 3 MB, generate random filenames, and block PHP execution inside uploads/.
- Escape user-provided content in HTML; validate CTA URLs and reject dangerous schemes such as javascript:.
- Generate a ZIP containing index.html, assets/ (CSS, JS, and uploaded images), and README.md. Test the archive on a clean standard LAMP host.
- Include database.sql for phpMyAdmin import and document database credentials, writable directories, PHP extensions, and ZIP support.

Implementation sequence

1. Scaffold authentication, dashboard, editor, export, and settings. **2.** Create the users/projects schema. **3.** Finish Glassmorphism and all editor fields. **4.** Add live preview and ZIP export. **5.** Complete localization and RTL. **6.** Add Aurora, Neubrutalism, Minimalist Dark, and 3D Gradient.

Acceptance target: a user can register, create and edit multiple projects, preview changes, upload safe images, switch language, and download a self-contained ZIP that runs on standard PHP/MySQL hosting.