ChatGPT

Saarzo Project Overview

Project Introduction

Saarzo Designs is a web-based platform for showcasing digital design templates (such as Instagram graphics, brand kits, and resume templates) and managing them through a simple admin interface. The site features a portfolio of creative templates for visitors to browse, and provides an administrative backend so a site owner can upload and organize these templates. In effect, Saarzo is a **template showcase and management system** built with standard web technologies. It presents a branded homepage and category-based template gallery to users, while enabling secure login and content management for the administrator.

Technologies Used

- **Frontend:** HTML, CSS, PHP – The user-facing pages (index.php, work.php, view.php, etc.) are built with standard HTML/CSS, with PHP embedded for dynamic content.
- **Backend:** PHP with a MySQL database – All data (template entries and contact messages) is stored in MySQL. PHP scripts handle inserting and retrieving records from the database, and manage user sessions.
- **Hosting:** InfinityFree – The site is deployed on a free InfinityFree hosting account, which offers full support for PHP and MySQL (e.g. PHP 8.3 and MySQL/MariaDB 8.0) [1]. InfinityFree provides a free subdomain and cPanel-style dashboard for file/DB management.
- **Firebase (legacy):** A previous version of Saarzo used Google Firebase (e.g. for authentication and realtime data), but the current deployment no longer uses it. (InfinityFree allows connecting to Firebase via its HTTP API [2], but all current data is handled via MySQL.)

Modules Covered

- **Landing Page (index.php):** The home page greets visitors with a prominent **hero section** showing the site title and tagline, plus an “Explore Now” button to go to the template gallery. It highlights a few featured template categories (Instagram pack, Brand kit, Resume, etc.) as cards or service listings. These template cards are displayed in a CSS-based grid/flex layout so that they wrap responsively on different screen sizes.
  Below the hero, the landing page dynamically loads featured template entries from the database and displays them as responsive cards. Each card shows a preview and links to the full template view. The layout is fully responsive: on smaller screens the header, hero text, and card grid reflow nicely (using CSS flexbox/grid), ensuring usability on mobile devices.

- **Template Display (work.php):** This page shows all templates or a filtered set based on category. It uses PHP to run a MySQL query to load template records (e.g. `SELECT * FROM portfolio_items`) and renders each as a card with an image and title. Category filters are implemented as links/buttons; for example, clicking an “Instagram” category reloads the page with a parameter (such as `work.php?category=Instagram`) and the PHP code runs a `SELECT * FROM portfolio_items WHERE category='Instagram'` query to filter results [3]. The page then displays only the templates in the chosen category.

- **Template View Page (view.php):** When a user clicks “View” on a template card, they are taken to `view.php?id={ID}`. The PHP code reads the `id` parameter, queries the database for that specific template (`SELECT * FROM portfolio_items WHERE id = {ID}`), and displays all its details: the title, description, category, and full-size image. This page presents the single-template information fetched from the database.

- **Admin Login (login.php):** A secure login form for the administrator. The page contains fields for username and password. On submission, PHP checks the credentials (hardcoded or from a database) and, if valid, starts a session and redirects the user to the dashboard; otherwise it shows an error. Session validation is used to protect admin pages. The login page features a stylized design (a full-screen gradient “trending” background) and the Saarzo logo. The form and button have custom CSS (e.g. gradients/hover styles) for a modern look.

- **Admin Dashboard (dashboard.php):** The main admin panel after login. It shows a navigation bar or header with links like “Dashboard”, “Messages”, “Edit Templates”, and “Logout”. The body of the page contains *cards or panels* for key actions: viewing user messages, uploading a new template, and editing/deleting existing templates. For example, there may be a “Messages” card linking to **message.php**, an “Upload” card to **uploads.php**, and an “Edit Templates” card to **edit.php**. These cards summarize the main admin functions and provide one-click access to those pages.

- **Upload Page (uploads.php):** A form page where the admin can add a new template. The form includes fields for *Title*, *Description*, *Category* (e.g. via a dropdown), and an image file upload control. When submitted, the PHP script handles the input: it moves the uploaded image file into the `/uploads/` directory on the server, and inserts a new record into the `portfolio_items` MySQL table with the template’s details (storing the title, description, category, and image filename). This immediately adds the new template so that it will appear on the user-facing gallery pages.

- **Edit Page (edit.php):** Provides functionality to update or delete existing templates. The page either lists templates in a table or is reached via the “Edit Templates” dashboard link. The admin can edit a template’s information (title, description, category, and image) and save changes, which updates the corresponding database record. There is also a delete option (e.g. a “Delete” link) for each template. Note: a bug fix was applied so that delete links match the correct file naming (for example, ensuring the URL `delete.php?id=X` matches the script filename exactly) to avoid HTTP 404 errors seen earlier.

- **Message View (message.php):** Shows contact form submissions from users. On the front-end `contact.php` page, visitors fill out a form with their *Name*, *Email*, and *Message*. When submitted, PHP saves these into a `messages` MySQL table. The `message.php` admin page queries that table and displays the list of messages in a table or cards (showing name, email, and message content). This lets the admin review and manage user inquiries.

UI/UX Enhancements

The Saarzo design uses modern CSS styling for a polished look. A custom circular logo is prominently placed in the header for consistent branding. The site employs **CSS gradients** (e.g. layered background colors) and **box-shadows** on cards and images to add depth. Buttons have custom colors and hover effects for interactivity. The login page uses a trendy full-screen gradient background. All pages use a responsive layout (CSS flexbox/grid) so that the header, footer, and template cards rearrange smoothly on different devices.

- Gradient and box-shadow styling on backgrounds, cards, and images (for a modern, layered look)
- Custom button designs with hover effects for better interaction feedback
- Circular Saarzo logo in the header (as shown above) to reinforce branding
- “Trending” style full-screen gradient background on the login page
- Responsive layout for card grids, header, and footer (CSS Grid/Flex ensures good display on mobile/tablet)

File Organization

- **Root directory:** Contains public pages and config. This includes `index.php` (home), `about.php`, `work.php` (template gallery), `view.php` (individual template page), `contact.php`, and `config.php` (database connection settings).
- **admin/** directory: Contains admin scripts – `login.php` (admin login form), `dashboard.php` (admin home), `uploads.php` (template upload form), `edit.php` (edit/delete templates), and `message.php` (view messages).
- **uploads/** directory: Holds media files. This includes `logo.jpg` (the Saarzo logo) and any template image files uploaded by the admin for display.

How Data Flows

- **Adding a Template:** When the admin uses the Upload page, the form data (title, desc, category) and image file are sent to a PHP script. The image is saved in `/uploads/` and a new row is inserted into the `portfolio_items` database table with the template’s data. After this, the template appears in the gallery.
- **Displaying Templates:** The templates gallery (`work.php`) runs a PHP `SELECT` query on the `portfolio_items` table to retrieve templates. It then loops over the results and generates HTML cards for each template.
- **Category Filtering:** If a user clicks on a specific category (e.g. “Instagram”), the request includes a category parameter. The PHP code then executes a filtered SQL query (e.g. `SELECT * FROM portfolio_items WHERE category='Instagram'`) to fetch only matching templates [3]. Only those templates are displayed.
- **Viewing a Template:** Clicking a template’s “View” button sends the browser to `view.php?id={template_id}`. The PHP code takes that `id` from the URL and runs `SELECT * FROM portfolio_items WHERE id={template_id}`. The single matching record is fetched and its details (title, desc, image) are displayed on the page.
- **Contact Messages:** The front-end contact form posts user data to a PHP script that inserts the name, email, and message into a `messages` database table. Later, when the admin visits **message.php**, PHP selects all rows from `messages` and displays them in a table or list, so the admin can read each user submission.

Hosting and Deployment

- **File Deployment:** The site is hosted on InfinityFree. As per the InfinityFree documentation, you deploy by uploading all PHP/HTML/CSS files into the hosting account’s `htdocs/` folder (either via the online File Manager or FTP) [4]. If the PHP code contains any base URLs or paths, update them for the new domain if needed.

- **Database Setup:** In the InfinityFree control panel, a MySQL database is created (InfinityFree allows up to 400 databases [5]). Using phpMyAdmin (provided by the host), the two tables (`portfolio_items` and `messages`) are created or imported with the proper schema. The InfinityFree credentials (hostname, username, password, database name) are then placed into `config.php`.
- **Configuration:** Ensure that `config.php` (or similar include) uses the InfinityFree database credentials. Any hard-coded domain or path settings should be updated to the InfinityFree subdomain (or custom domain) in the code. After upload and DB import, the site runs live at the provided URL.

Legacy Firebase Integration (Optional)

Earlier in development, Saarzo used Google Firebase services for certain features (for example, user login and/or real-time data). InfinityFree does **allow** including Firebase SDKs and communicating with Firebase APIs [2], but the *currently deployed version* of Saarzo no longer uses Firebase. All authentication and data storage is now handled via PHP sessions and the MySQL database.

Sources: The above summary is based on the Saarzo project files and design (as described) together with hosting documentation. For example, InfinityFree's hosting features include full PHP & MySQL support [1] and advise uploading all files to `htdocs` and importing the database via phpMyAdmin [4]. The category filtering SQL example is analogous to community solutions (see e.g. a `SELECT ... WHERE category=...` example [3]). The Firebase note is based on platform compatibility (InfinityFree admin forum confirms Firebase can be used [2]), even though Saarzo now uses MySQL. All cited references reflect platform documentation and common PHP/MySQL practices.

---

[1] [5] Free Web Hosting with PHP and MySQL - InfinityFree
https://www.infinityfree.com/

[2] Can I use firebase in website hosting on your platform? - Hosting Support - InfinityFree Forum
https://forum.infinityfree.com/t/can-i-use-firebase-in-website-hosting-on-your-platform/73437

[3] mysql - Trying to create category wise filtering in the view where i list stuff and getting an error in php? - Stack Overflow
https://stackoverflow.com/questions/1389213/trying-to-create-category-wise-filtering-in-the-view-where-i-list-stuff-and-gett

[4] Bringing website to InfinityFree - Hosting Support - InfinityFree Forum
https://forum.infinityfree.com/t/bringing-website-to-infinityfree/78645