

Rapport projet tutoré : interface de générateur de site statique

Contexte :

Avec le développement du web et la diversification des besoins en communication numérique, de plus en plus d'organisations cherchent à créer des sites web simples, rapides et sécurisés.

Les générateurs de sites statiques répondent parfaitement à ces attentes : ils produisent des sites composés uniquement de fichiers HTML, sans base de données, ce qui leur confère performance, stabilité et sécurité.

Mais ces générateurs sont destinés à un public ayant des connaissances techniques puisqu'il nécessite la maîtrise de :

- de la ligne de commande
- de formats de configuration tels que TOML / YAML
- de systèmes de templates comme Tera
- de la structuration stricte des dossiers du projet

Cette complexité représente un frein pour les utilisateurs débutants ou non techniques qui souhaitent simplement créer un site sans apprendre des outils de développement.

Malgré ses nombreux avantages (vitesse, simplicité du rendu, open-source), les générateurs de site comme Zola ne sont pas facilement accessibles aux personnes ne maîtrisant pas les concepts techniques du web.

Problématique :

Comment permettre à des utilisateurs non techniques de bénéficier des avantages des générateurs de sites statiques, tout en supprimant les barrières liées aux outils de développement ?

Objectif :

Il existe donc un besoin réel d'outiller et de simplifier l'utilisation de Zola, en proposant une interface graphique qui :

- cache la complexité technique
- automatise la création et l'organisation des contenus
- permet de générer un site statique en quelques clics
- facilite la prise en main pour tout public

Notre projet a donc pour but de rendre l'édition de site web plus intuitive et accessible en créant une interface simple d'utilisation.

Liste fonctionnalité :

- création d'un site avec un formulaire
 - url
 - bouton switch pour chaque possibilité de paramétrage avec text explicatif
- Formulaire de création de fichier markdown
 - titre
 - date
 - bouton ajout de text
 - bouton pour formater le texte (gras, h1, lien...)
 - bouton ajouter photo et/ou vidéo (ajouter l'image dans le /static/images et ajouter en md le code)
- sélecteur de thème
 - choix de la positions des éléments
 - choix de la couleurs
- Bouton déploiement du serveur
- Gestion de compte (connexion inscription)
- Visualisation du site en temps réel
- Génération du site final

Périmètre du projet :

- Le projet se concentre sur l'interface graphique et l'automatisation des tâches principales de Zola.
- Il ne vise pas à remplacer complètement Zola mais à en simplifier l'usage.
- Le déploiement est pris en charge via des services existants (GitHub Pages, Netlify).

Limites :

- Personnalisation avancée limitée pour les utilisateurs non administrateurs.
- Pas de génération de sites dynamiques (PHP, base de données).
- Interface pensée pour un usage pédagogique ou pour des petits/moyens projets.

Sécurité et gestion des accès :

- Authentification via compte utilisateur
- Rôles (utilisateur / administrateur)
- Protection de l'accès aux fichiers système
- Validation des fichiers envoyés (images, vidéos)

Acteur :

Utilisateur	Crée, édite et organise les pages/sections via l'interface. Génère le site statique.
Administrateur	Peut personnaliser les templates et configurer les paramètres avancés.
Générateur de site	Système externe appelé par l'interface pour générer les fichiers HTML.
Système de fichier	Stocke les fichiers Markdown, templates, images et site généré.

Cas d'usage de l'utilisateur :

Créer une nouvelle page :

- **Description** : Générer un fichier Markdown avec un titre et un contenu en utilisant un template existant
- **Objectif** : Ajouter du contenu au site

Modifier le contenu d'une page existante :

- **Description** : Éditer le texte ou les métadonnées d'une page existante
- **Objectif** : Mettre à jour le contenu

Supprimer une page :

- **Description** : Retirer une page inutilisée du projet
- **Objectif** : Maintenir l'organisation du site

Ajouter une section :

- **Description** : Ajouter une nouvelle section contenant un `_index.md`
- **Objectif** : Organiser les pages par thème

Gérer la navigation :

- **Description** : Ajouter ou modifier les éléments du menu
- **Objectif** : Améliorer la structure navigationnelle

Ajouter des images ou fichiers :

- **Description** : Importer des ressources dans `static/` et les intégrer
- **Objectif** : Enrichir les contenus

Prévisualiser le site :

- **Description** : Afficher une version locale du site statique généré
- **Objectif** : Vérifier le rendu avant publication

Générer le site statique :

- **Description** : Compiler les fichiers Markdown en site HTML statique
- **Objectif** : Produire le site prêt au déploiement

Cas d'usage de l'administrateur :

Configurer les paramètres du générateur :

- **Description** : Modifier la configuration globale du site
- **Objectif** : Adapter le site aux besoins du projet

Installer / gérer les thèmes:

- **Description** : Ajouter, supprimer ou activer un thème Zola
- **Objectif** : Changer l'apparence globale du site

Créer / modifier des templates :

- **Description** : Éditer les fichiers HTML/Tera dans *template/*
- **Objectif** : Gérer la structure du site

Gérer les droits utilisateurs :

- **Description** : Définir qui peut créer/éditer des pages
- **Objectif** : Sécuriser l'accès à l'interface

Gérer les ressources statiques :

- **Description** : Ajouter ou organiser les ressources (*static/*) : JSON, IMAGE, FONT
- **Objectif** : Assurer l'intégrité des assets

Configurer le déploiement :

- **Description** : Définir où et comment le site est publié (Netlify, GitHub Pages...)
- **Objectif** : Automatiser la mise en ligne

Autoriser et lancer le déploiement :

- **Description** : Vérification finale du site avant publication
- **Objectif** : Vérifier que l'on poste pas n'importe quoi

Spécification technique

Architecture

- **Application de bureau (Desktop)** : Développée avec **Electron**, intégrant le moteur de rendu web (Chromium) et l'environnement d'exécution (Node.js) dans un seul exécutable.
- **Système de Fichiers (Local)** : L'application communique directement avec les fichiers de l'utilisateur via le module fs de Node.js. Elle ne nécessite ni serveur distant, ni API RESTful, ni base de données.
- **Moteur Zola** : L'application pilote directement l'exécutable binaire zola (via child_process) pour lancer le serveur de prévisualisation et générer le site final.

Interface

- **Éditeur Dynamique** : L'interface génère automatiquement les champs du formulaire (Titre, Date, Booléens, Tags...) en analysant les métadonnées (Front Matter TOML/YAML) du fichier Markdown sélectionné.
- **Gestion de Projet** : Chargement de projets locaux via une boîte de dialogue système native, permettant de travailler sur n'importe quel dossier existant sur la machine.
- **Prévisualisation** : Contrôle direct du serveur Zola (zola serve) avec retour visuel des erreurs dans l'interface.

Maquettes Figma:

<https://www.figma.com/design/MA5m0gcGGaWpR4tBOcz9u3/Projet-tutor%C3%A9-interface-web?node-id=0-1&p=f&t=YVowBufJkE5HwFAQ-0>