

Cahier des charges - Site Web Thématique

- Yasmina IMLYHEN - EEDN -

Évaluation 1

30 Juin 2025

TABLE DES MATIÈRES

1. Présentation du projet

- Objectif
- Thèmes au choix
- Option 2 - Thème libre
- Public cible

2. Contraintes & contraintes techniques

- Gestion du projet
- Maquette responsive
- Structure HTML
- Styles CSS
- JavaScript
- Accessibilité
- Fonctionnalités visuelles

3. Spécifications fonctionnelles détaillées

- Page d'accueil (index.html)
- Fonctionnalités JavaScript
- Page Contenu (content.html)
- Page Contact (contact.html)
- Fonctionnalités JavaScript

4. Structure des données JSON (adaptée au thème) (Option)

5. Architecture JavaScript

- utilities.js
- manager.js
- app.js

6. Spécifications techniques détaillées

- Responsive Design
- Performance
- Accessibilité (WCAG 2.1 AA - Essentiel)

7. Livrables attendus

- Fichiers sources
- Documentation
- Maquette responsive

8. Critères d'évaluation

- Technique (40%)
- Fonctionnel (30%)
- Design et Accessibilité (30%)

9. Bonus (points supplémentaires)

10. Contraintes de temps

Cahier des charges - Site Web Thématique

1. Présentation du projet

Objectif :

Développer un site web thématique responsive avec des fonctionnalités interactives modernes et une expérience utilisateur optimale.

Thèmes au choix :

—> Option 1 - Blog Tech :

- Blog d'actualités technologiques
- Articles sur les nouvelles technologies
- Tutorials et guides techniques
- Communauté de développeurs

—> Option 2 - Thème libre :

- Restaurant et gastronomie
- Agence de voyage
- Boutique en ligne (vitrine)
- Portfolio artistique
- Site associatif
- Magazine lifestyle
- Plateforme éducative
- Ou tout autre thème de votre choix

Public cible :

Adapté au thème choisi (lecteurs tech, clients potentiels, visiteurs intéressés, etc.)

2. Contraintes & contraintes techniques

Maquette responsive :

- Maquette responsive: une maquette (wireframe ou prototype) du site doit être réalisée avant le développement, pour mobile, tablette et desktop.
- Développement basé sur la maquette : le site doit respecter la maquette validée.

Gestion du projet :

- sur GitHub:
- le code source doit être versionné sur un dépôt GitHub public, accessible à l'évaluateur.

Structure HTML :

Minimum 3 pages HTML :

- index.html (page d'accueil)
- content.html (articles/produits/réalisations selon le thème)
- contact.html (formulaire et informations)

Styles CSS :

Minimum 2 feuilles de style :

- styles.css (styles principaux)
- responsive.css (media queries et adaptations)
- Utilisation obligatoire de Flexbox pour tous les layouts
- Design responsive adapté mobile, tablette, desktop
- Respect des normes W3C et validation HTML/CSS

JavaScript :

3 fichiers JS obligatoires :

- utilities.js (fonctions utilitaires réutilisables)
- manager.js (gestion des données et états)
- app.js (logique principale et initialisation)
- Manipulation du DOM native (sans frameworks)
- Gestion des événements (clicks, scroll, resize, etc.)
- Objets JavaScript simples {} pour structurer les données
- Utilisation de JSON pour les données de contenu (option)
- LocalStorage OU SessionStorage pour la persistance (option)

Accessibilité :

- Conformité WCAG 2.1 niveau AA
- Attributs ARIA appropriés
- Navigation au clavier
- Contrastes de couleurs respectés
- Textes alternatifs pour les images
- Structure sémantique HTML5

Fonctionnalités visuelles :

- Slider d'images OU effet parallax (au choix)
- Animations CSS et/ou JavaScript fluides

3. Spécifications fonctionnelles détaillées**Page d'accueil (index.html)*****Contenu requis :***

- Header avec navigation simple

- Section hero avec présentation du site/thème
- Section mise en avant (articles récents, produits phares, etc.)
- Slider simple d'images OU effet parallax léger
- Footer avec informations essentielles

Fonctionnalités JavaScript :

- Menu hamburger responsive
- Slider manuel simple (si choisi)
- Smooth scrolling vers sections

Page Contenu (content.html)

Exemples selon le thème :

-> Si Blog Tech :

- Liste d'articles techniques
- Système de catégories (Frontend, Backend, Mobile, etc.)
- Modal de lecture d'articles
- Système de favoris d'articles

-> Si thème libre (ex: Restaurant) :

- Galerie de plats/menus
- Catégories (Entrées, Plats, Desserts)
- Modal de détail des plats
- Système de favoris/wishlist

Fonctionnalités JavaScript communes:

- Ouverture/fermeture de modals
- Sauvegarde des favoris en LocalStorage
- Affichage dynamique du contenu

Page Contact (contact.html)

Contenu requis :

- Formulaire de contact simple (nom, email, message)
- Informations de contact basiques
- Historique simplifié des messages

Fonctionnalités JavaScript :

- Validation basique du formulaire
- Sauvegarde des messages en SessionStorage
- Affichage de l'historique
- Confirmation d'envoi simple

4. Structure des données JSON (adaptée au thème) (Option)

Exemple Blog Tech (articles.json)

```
``json
{
  "articles": [
    {
      "id": 1,
      "title": "Les nouveautés JavaScript 2025",
      "category": "frontend",
      "excerpt": "Découvrez les dernières fonctionnalités...",
      "author": "Jane Doe",
      "date": "2025-06-01",
      "image": "js2025.jpg",
      "tags": ["JavaScript", "ES2025"]
    },
    {
      "id": 2,
      "title": "Guide Node.js pour débutants",
      "category": "backend",
      "excerpt": "Apprenez les bases de Node.js...",
      "author": "John Smith",
      "date": "2025-05-28",
```



```
    "image": "nodejs.jpg",
    "tags": ["Node.js", "Backend"]
  }
]
}
...
```

Exemple thème libre - Restaurant (menu.json)

```
```json
{
 "dishes": [
 {
 "id": 1,
 "name": "Salade César",
 "category": "entrees",
 "description": "Salade fraîche avec croûtons...",
 "price": "12€",
 "image": "cesar.jpg",
 "allergens": ["gluten", "œufs"]
 },
 {
 "id": 2,
 "name": "Saumon grillé",
 "category": "plats",
 "description": "Filet de saumon avec légumes...",
 "price": "22€",
 "image": "saumon.jpg",
 "allergens": ["poisson"]
 }
]
}
...
```

### ***Données de configuration (config.json)***

```
```json
{
  "site": {
    "title": "TechBlog 2025",
    "description": "Votre source d'actualités tech",
    "contact": {
      "email": "contact@example.com",
      "phone": "01 23 45 67 89"
    }
  }
}
```

```
}  
}  
}  
...
```

5. Architecture JavaScript

utilities.js

Fonctions essentielles :

- createElement(tag, className, content)
- toggleClass(element, className)
- validateEmail(email)
- formatDate(date)

manager.js

Gestionnaires essentiels :

- DataManager: chargement JSON simple (option)
- StorageManager : interface LocalStorage/SessionStorage (option)
- ModalManager : ouverture/fermeture modals (si choisi)
- SliderManager : gestion slider basique (si choisi)

app.js

Initialisation simplifiée :

- Initialisation des managers
- Binding des événements principaux
- Logique de navigation simple

6. Spécifications techniques détaillées

Responsive Design

—> Breakpoints

- **Mobile** : 320px - 767px
- **Tablette** : 768px - 1023px
- **Desktop** : 1024px+
- **Large Desktop** : 1200px+

Performance

- Images optimisées (formats web standards)
- Code propre et commenté
- Chargement rapide priorité sur les optimisations avancées

Accessibilité (WCAG 2.1 AA - Essentiel)

- Navigation clavier sur éléments interactifs
- Contraste minimum respecté
- Alt texts sur images importantes
- Structure sémantique HTML5 basique

—> *Quelques attributs*

- alt : Descriptions alternatives pour images
- aria-label : Labels pour éléments interactifs
- aria-labelledby : Référence à un élément de titre
- aria-describedby : Descriptions détaillées
- aria-expanded : État des éléments extensibles
- aria-hidden : Masquage d'éléments décoratifs
- aria-live : Annonces dynamiques (polite/assertive)
- aria-current : Indication de l'élément actuel
- role : Définition du rôle sémantique

- tabindex : Gestion de l'ordre de navigation
- for : Association label/input
- title : Informations contextuelles

7. Livrables attendus

IMPORTANT Votre dossier doit être nommé : eval1_prenom_nom

Fichiers sources

```
projet/
├── index.html
├── content.html (ou blog.html, menu.html, etc.)
├── contact.html
├── css/
│   ├── styles.css
│   └── responsive.css
├── js/
│   ├── utilities.js
│   ├── manager.js
│   └── app.js
├── data/
│   ├── content.json (articles.json, menu.json, etc.)
│   └── config.json
├── images/
└── README.md
```

Documentation

- README.md
- Commentaires détaillés dans le code

Maquette responsive

- Maquette responsive (fichier Figma, Adobe XD, ou PDF, etc.)
- Lien du dépôt GitHub public contenant l'intégralité du projet et l'historique des commits

8. Critères d'évaluation

Technique (40%)

- Respect des contraintes HTML/CSS/JS
- Qualité et propreté du code
- Architecture JavaScript modulaire
- Validation W3C

Fonctionnel (30%)

- Fonctionnalités implémentées et opérationnelles
- Gestion des erreurs et cas limites
- Persistance des données
- UX fluide

Design et Accessibilité (30%)

- Design responsive et cohérent
- Respect des normes d'accessibilité
- Animations fluides et appropriées
- Expérience utilisateur optimale

9. Bonus (points supplémentaires)

- Animations CSS simples mais efficaces

- Dark/Light mode avec sauvegarde

10. Contraintes de temps

- Durée obligatoire : 5 jours maximum

—> Proposition planning :

- Jour 1 : Structure HTML des 3 pages + CSS de base avec Flexbox
- Jour 2 : JavaScript core (utilities.js + manager.js) + fonctionnalités principales
- Jour 3 : Finalisation des interactions, slider/parallax, responsive
- Jour 4 : Tests, optimisations, accessibilité et documentation
- Jour 5 : Revue complète et livraison

Note importante:

Ce projet est conçu pour être réalisable en 5 jours.

Vous êtes libre de choisir le thème qui vous inspire le plus (blog tech recommandé pour la cohérence, ou thème libre pour la créativité).

L'accent est mis sur le respect des contraintes techniques imposées (Flexbox, 3 fichiers JS ...) plutôt que sur la complexité des fonctionnalités. La priorité est donnée à un code propre, fonctionnel et respectueux des standards.