



Ce cahier des charges est fourni par le laboratoire indépendant **SynaptikOne**, à titre pédagogique. Il peut être utilisé et adapté pour des projets personnels ou universitaires, sous réserve du respect des conditions de la licence suivante.

Licence : Creative Commons BY-NC-SA 4.0 — Usage commercial interdit.

Site web : <https://synaptikone.pages.dev/>

Facebook : Page Facebook SynaptikOne

Cahier des Charges

Plateforme E-Learning (Front-End Only)

Rakotondravelo Tahina Mickaël

Fondateur de SynaptikOne

November 28, 2025

Contents

1	Présentation générale du projet	4
1.1	Contexte	4
1.2	Objectif principal	4
2	Objectifs pédagogiques	4
3	Périmètre fonctionnel	5
3.1	Pages à réaliser	5
3.1.1	Page d'accueil	5
3.1.2	Page liste des cours	5
3.1.3	Page d'un cours	5
3.1.4	Page quiz	6
3.1.5	Page profil (optionnel)	6
4	Fonctionnalités JavaScript	6
4.1	Recherche dynamique	6
4.2	Filtres combinés	6
4.3	Suivi de progression	6
4.4	Quiz interactif	7
4.5	Mode sombre	7
4.6	Animations (optionnel)	7

5	Design et expérience utilisateur	7
5.1	Identité visuelle	7
5.2	Mise en page	7
5.3	Composants UI	7
6	Contraintes techniques	8
7	Organisation du projet	8
7.1	Structure de fichiers recommandée	8
7.2	Travail en équipe (optionnel)	8
8	Livrables attendus	9
9	Critères d'évaluation	9
10	Évolutions possibles (bonus)	9

1 Présentation générale du projet

1.1 Contexte

Dans le cadre du projet pédagogique, les étudiants doivent concevoir et réaliser une plateforme d'apprentissage en ligne accessible sur tout support (ordinateur, tablette, mobile). Le système doit fonctionner uniquement en **Front-End**, c'est-à-dire avec HTML, CSS et JavaScript, sans backend ni base de données serveur.

1.2 Objectif principal

L'objectif est de créer une application web moderne permettant aux utilisateurs de :

- consulter une liste de cours,
- accéder à un cours détaillé (vidéo, chapitres),
- suivre leur progression,
- passer un quiz interactif,
- utiliser une interface ergonomique et responsive.

2 Objectifs pédagogiques

- Maîtriser HTML5 avancé.
- Créer un design professionnel avec CSS3 (Flexbox, Grid, responsive).
- Manipuler JavaScript pour :
 - interactions visuelles,
 - filtres dynamiques,
 - quiz interactifs,
 - stockage local des données (LocalStorage).
- Organiser un projet complet multi-pages.
- Adopter une approche modulaire avec fichiers séparés.

3 Périmètre fonctionnel

3.1 Pages à réaliser

3.1.1 Page d'accueil

Contient :

- logo + menu,
- bannière principale,
- slogan,
- barre de recherche,
- catégories de cours,
- section des cours populaires,
- pied de page.

3.1.2 Page liste des cours

Fonctionnalités :

- recherche instantanée,
- filtres (catégorie, niveau),
- affichage en cartes,
- chaque carte contient un visuel, un titre, un niveau, un bouton.

3.1.3 Page d'un cours

- titre du cours,
- description,
- vidéo intégrée,
- liste des chapitres,
- barre de progression,
- bouton *Passer au quiz*.

3.1.4 Page quiz

- questions en QCM,
- validation des réponses,
- score final,
- sauvegarde du score.

3.1.5 Page profil (optionnel)

- statistiques utilisateur,
- progression des cours,
- mode sombre.

4 Fonctionnalités JavaScript

4.1 Recherche dynamique

Filtrage en temps réel de la liste des cours selon un mot-clé.

4.2 Filtres combinés

Filtres par :

- catégorie,
- niveau,
- mot-clé.

Les filtres doivent fonctionner simultanément.

4.3 Suivi de progression

Utilisation de LocalStorage pour enregistrer :

- le chapitre courant,
- le pourcentage de progression,
- le score du quiz.

4.4 Quiz interactif

- questions stockées en JSON ou fichier JS,
- correction dynamique,
- score final affiché.

4.5 Mode sombre

Basculer entre deux thèmes persistants grâce à LocalStorage.

4.6 Animations (optionnel)

Possibilité d'utiliser AOS ou GSAP pour des animations modernes.

5 Design et expérience utilisateur

5.1 Identité visuelle

- palette cohérente et moderne,
- typographies Google Fonts,
- design minimaliste.

5.2 Mise en page

- utilisation de Flexbox et CSS Grid,
- responsive : mobile, tablette, desktop.

5.3 Composants UI

- cartes,
- boutons stylés,
- barres de progression,
- navigation fixe ou mobile.

6 Contraintes techniques

- HTML5, CSS3, JavaScript uniquement,
- aucun backend (pas de PHP, SQL, Node.js),
- données stockées en fichiers JSON ou JS,
- utilisation du LocalStorage pour simuler une base de données,
- déploiement possible sur GitHub Pages ou Netlify.

7 Organisation du projet

7.1 Structure de fichiers recommandée

```
/assets  
/css  
style.css  
responsive.css  
/js  
app.js  
courses.js  
quiz.js  
progress.js  
/img  
index.html  
courses.html  
course-view.html  
quiz.html  
profile.html
```

7.2 Travail en équipe (optionnel)

- chef de projet,
- designer UI/UX,
- intégrateur HTML/CSS,
- développeur JavaScript.

8 Livrables attendus

1. site complet fonctionnel,
2. code source propre et organisé,
3. documentation PDF comprenant :
 - description du projet,
 - maquettes,
 - architecture du code,
 - fonctionnalités,
 - difficultés rencontrées,
 - solutions proposées.
4. présentation orale avec démonstration.

9 Critères d'évaluation

- richesse des fonctionnalités,
- qualité visuelle,
- responsive design,
- clarté et organisation du code,
- maîtrise de JavaScript,
- originalité ajoutée,
- qualité de la présentation finale.

10 Évolutions possibles (bonus)

- tableau de bord administrateur simulé,
- système de badges,
- utilisation d'une API externe,
- notation des cours,
- commentaires simulés.

Arborescence du projet

```
e-learning/
|
|-- index.html          -> Page d'accueil
|-- courses.html        -> Liste de tous les cours
|-- course-view.html    -> Page d'un cours
|-- quiz.html           -> Page du quiz
|-- profile.html        -> Profil utilisateur (optionnel)
|
|-- assets/
|   |-- css/
|   |   |-- style.css    -> Styles généraux
|   |   |-- layout.css   -> Structure (grilles, flex)
|   |   |-- components.css -> Cartes, boutons, menus...
|   |   |-- theme.css    -> Mode clair/sombre
|   |   \-- responsive.css -> Adaptation mobile/tablette
|   |
|   |-- js/
|   |   |-- app.js       -> Script global
|   |   |-- data/
|   |   |   |-- courses.js -> Base locale des cours
|   |   |   \-- quiz-data.js -> Questions du quiz
|   |   |-- courses.js    -> Filtres + recherche
|   |   |-- course-view.js -> Gestion affichage d'un cours
|   |   |-- quiz.js       -> Logique du quiz
|   |   \-- progress.js   -> Progression (LocalStorage)
|   |
|   |-- img/
|   |   |-- logo/
|   |   |-- courses/
|   |   |-- icons/
|   |   \-- ui/
|   |
|   |-- fonts/
|   |
|   \-- libs/
```

```
|      |-- aos/  
|      |-- gsap/  
|      |-- chartjs/  
|      \-- bootstrap/  
|  
|-- data/  
|  \-- courses.json  
|  
\-- README.md
```