

Cahier des charges – Projet Chrono Explorer







1. Contexte

Le projet **Chrono Explorer** est une application web interactive permettant d'explorer les grands événements de l'histoire. Elle propose une navigation simple à travers une ligne du temps, un système de favoris, des commentaires et un espace d'administration.

2. Objectifs

- Proposer une expérience immersive et pédagogique autour de l'histoire
- Permettre aux utilisateurs de consulter des événements, d'interagir et de les sauvegarder
- Offrir une interface simple, claire, responsive et moderne
- Mettre en place une zone de modération réservée à l'administrateur
- Gérer l'authentification de manière simple via le navigateur

3. Fonctionnalités principales

Module	Description
 Accueil	Ligne du temps avec événements historiques
 Détail événement	Fiche complète avec image, date, description, médias
 Favoris	Ajout/suppression de favoris (par utilisateur)
 Authentification	Inscription, connexion, déconnexion (localStorage)
 Modération admin	Validation/suppression de commentaires
 Carte interactive (optionnelle)	Affiche la localisation des événements

4. Technologies utilisées

Technologie	Détail
Framework Frontend	Angular 19.2.11 (standalone components)
Langages	TypeScript (Angular & Node), HTML, SCSS
Base de données	MySQL (via phpMyAdmin ou PlanetScale)
Backend / API	Node.js + Express – API REST
Stockage temporaire	LocalStorage (auth, favoris par utilisateur)
UI	Bootstrap 5, SCSS personnalisé
Librairies	Leaflet.js (cartes), RxJS
Outils de build	Angular CLI, npm, tsconfig
Hébergement cible	Vercel (front), Render (API), PlanetScale (BDD)

5. Architecture du projet

Le projet repose sur une **architecture full-stack** répartie en trois couches principales :

- **Frontend** : développé avec Angular 19 en standalone components
- **Backend** : développé en **Node.js + Express**, structuré en microservices REST (événements, médias, utilisateurs)
- **Base de données** : système **MySQL** (phpMyAdmin), avec tables pour événements, utilisateurs, médias, et commentaires

La communication se fait via des **API REST** sécurisées, et les utilisateurs sont authentifiés par des endpoints dédiés. L'application est modulaire et respecte les bonnes pratiques de séparation des responsabilités.

6. Relations et données

- 1 événement peut avoir :
 - plusieurs **médias**
 - plusieurs **commentaires**
- Pas de base de données relationnelle formelle, mais organisation logique claire
- Pas de diagramme UML ou MCD généré, mais les structures respectent une modélisation classique

7. Maquettes et design

L'interface a été conçue directement dans Angular sans outil formel de maquette, mais en respectant une logique claire, sobre, et responsive adaptée au mobile. L'inspiration provient de standards modernes et de composants Bootstrap.

8. Gestion du temps et planification

Le développement a été réparti sur plusieurs semaines :

- Semaine 1 -2 : installation, composants de base (accueil, événements)
- Semaine 3-4 : gestion des favoris, auth, commentaires
- Semaine 5-6 : modération admin, dark mode, ajustements
- Semaine 7-8 : intégration Leaflet, nettoyage final, contenu

Le projet a été livré dans les **délais impartis** malgré les ajustements et ajouts de fonctionnalités en cours de route. Une organisation agile (itérative) a permis une bonne adaptation.

9. Contraintes

- Le projet repose sur une architecture **full-stack**, avec un frontend Angular, un backend Node.js/Express, et une base de données MySQLResponsive mobile (testé sur différentes tailles d'écran)
- Doit fonctionner hors-ligne
- Aucun backend requis (usage de données locales)

10. Hébergement et écoconception

- Le projet peut être hébergé sur une plateforme type **Vercel**, compatible Angular
- Une attention particulière a été portée à l'impact environnemental :
 - Pas de surcharge de ressources
 - Application légère
 - Possibilité d'utiliser un hébergement "vert"

11. Livrables

- 📁 Code source Angular (/chrono-explorer)
- 📄 Plan marketing (plan-marketing.pdf)
- 📄 Cahier des charges (cahier-des-charges.pdf)
- (Facultatif) Lien Vercel ou dépôt GitHub

Conclusion

Le projet *Chrono Explorer* est une preuve de concept fonctionnelle, conforme aux exigences du sujet et enrichie de plusieurs fonctionnalités (favoris par utilisateur, modération, dark mode, carte). Il a été mené avec rigueur, dans les temps, et offre une base solide pour évoluer vers un vrai projet full-stack si nécessaire.