

README - Un auteur dans sa classe · Compte à rebours Salon du Livre 2026

Table des matières

- [Présentation du projet](#)
- [Types d'activités](#)
- [État d'avancement](#)
- [Architecture technique](#)
- [Prérequis](#)
- [Installation](#)
- [Configuration](#)
- [Structure du projet](#)
- [Guide de développement](#)
- [Gestion des activités](#)
- [Mode développement](#)
- [Déploiement](#)
- [Contribution](#)
- [Troubleshooting](#)

Présentation du projet

Vue d'ensemble

Application web interactive "**Un auteur dans sa classe**" permettant aux classes de maternelle à CM2 de suivre un **compte à rebours de 100 jours** avant un salon du livre jeunesse. Chaque jour d'école (lundi, mardi, jeudi, vendredi), une activité courte (≈10 minutes) est proposée pour découvrir les auteurs et illustrateurs invités.

Informations clés

- **URL de production** : <https://micetf.fr/un-auteur-dans-sa-classe>
- **Période** : Du J-100 (15 janvier 2026) au J-1 (24 avril 2026)
- **Événement** : Salon du livre jeunesse le 25 avril 2026
- **Nombre d'activités** : 48 activités réparties sur 4 mois
- **Public** : Maternelle, Cycle 2, Cycle 3
- **Usage** : Collectif, projeté (vidéoprojecteur/TNI)

Objectifs pédagogiques

- Installer un rituel quotidien de découverte culturelle
- Développer l'observation, le langage oral et la justification
- Créer de l'envie et de l'attente pour le salon du livre
- Proposer des activités adaptables de la maternelle au CM2
- Faire découvrir 13 auteurs et illustrateurs de littérature jeunesse

Auteurs invités

1. **Lionel TARCHALA** - Illustrateur
2. **Xavière BRONCARD** - Illustratrice
3. **Clémentine VAQUEZ** - Illustratrice
4. **Philippe JALBERT** - Auteur/Illustrateur
5. **Nathalie BONBON** - Autrice
6. **Nathalie DIETERLE** - Illustratrice
7. **Marion JACOUD** - Illustratrice
8. **Hervé LE GOFF** - Illustrateur
9. **Cécile POLLET** - Autrice/Illustratrice
10. **Séverine LAFOND** - Autrice
11. **Charlotte DESPLANCHE** - Autrice
12. **Christelle DUCHESNE** - Autrice
13. **Susie MORGENSTERN** - Autrice

Types d'activités

L'application propose **4 types d'activités** différentes pour varier les approches pédagogiques et maintenir l'engagement des élèves.

1. Quiz visuel

Objectif : Observer une illustration et répondre à une question

Fonctionnement :

- Une grande image de l'auteur/illustrateur est affichée à gauche (60% de l'écran)
- 3 propositions de réponse sont présentées à droite (40% de l'écran)
- Les élèves discutent collectivement puis votent
- L'enseignant clique sur la proposition choisie
- Un feedback immédiat apparaît avec une explication pédagogique

Format JSON :

```
{
  "type": "quiz",
  "image": "/images/activites/2026-01-15/image-principale.jpg",
  "consigneEleves": "Quel animal vois-tu sur cette illustration ?",
  "propositions": [
    {
      "id": "a",
      "texte": "Un dragon",
      "image": "/images/activites/2026-01-15/proposition-a.jpg",
      "correct": true,
      "commentaire": "Bravo ! C'est bien un dragon..."
    },
    {
      "id": "b",
      "texte": "Un chat",
```



```

        "correct": false,
        "commentaire": "Non, observe bien les écailles..."
    },
    {
        "id": "c",
        "texte": "Un oiseau",
        "correct": false,
        "commentaire": "Les ailes ressemblent, mais regarde la
queue..."
    }
]
}

```

Variantes pédagogiques :

- Questions d'observation : "Que vois-tu ?"
- Questions de compréhension : "Où se passe la scène ?"
- Questions d'interprétation : "Quelle émotion ressens-tu ?"
- Questions techniques : "Quelle technique a été utilisée ?"

Images nécessaires :

- [image-principale.jpg](#) : L'illustration à observer (1200-1600px)
- [proposition-a.jpg](#) : Image optionnelle pour la proposition A (600-800px)
- [proposition-b.jpg](#) : Image optionnelle pour la proposition B
- [proposition-c.jpg](#) : Image optionnelle pour la proposition C

État : ✓ **IMPLÉMENTÉ**

2. Jeu de l'intrus 🔍

Objectif : Identifier l'élément qui ne correspond pas aux autres

Fonctionnement :

- 3 ou 4 vignettes sont affichées en grille (adaptative selon le nombre)
- Chaque vignette occupe un grand espace (33% ou 50% de l'écran)
- Les élèves observent et discutent collectivement
- Ils doivent identifier l'intrus et justifier leur choix
- L'enseignant clique sur la vignette choisie
- L'intrus s'agrandit avec un badge 🗯️ et une explication apparaît

Format JSON :

```

{
  "type": "intrus",
  "consigneEleves": "Quelle image est l'intrus ?",
  "vignettes": [
    {
      "id": "1",
      "image": "/images/activites/2026-01-16/vignette-1.jpg",

```



```

        "titre": "Image 1",
        "isIntrus": false,
        "explication": "Cette image partage le même style que les
autres."
    },
    {
        "id": "2",
        "image": "/images/activites/2026-01-16/vignette-2.jpg",
        "titre": "Image 2",
        "isIntrus": true,
        "explication": "Bravo ! C'est l'intrus car c'est la seule en
noir et blanc."
    },
    {
        "id": "3",
        "image": "/images/activites/2026-01-16/vignette-3.jpg",
        "titre": "Image 3",
        "isIntrus": false,
        "explication": "Celle-ci utilise les mêmes couleurs vives."
    }
]
}

```

Variantes pédagogiques :

- Intrus par technique : aquarelle vs numérique
- Intrus par style : réaliste vs fantastique
- Intrus par couleur : couleurs chaudes vs froides
- Intrus par composition : pleine page vs cases
- Intrus par personnage : personnage récurrent

Images nécessaires :

- [vignette-1.jpg](#) : Première vignette (600-800px carré)
- [vignette-2.jpg](#) : Deuxième vignette
- [vignette-3.jpg](#) : Troisième vignette
- [vignette-4.jpg](#) : Quatrième vignette (optionnel)

État : ✓ IMPLÉMENTÉ

3. "Je lis une image" 📖

Objectif : Analyser une illustration en profondeur avec plusieurs questions successives

Fonctionnement :

- Une grande illustration occupe tout l'écran
- Une série de 2-3 questions sont posées successivement
- Les élèves répondent à chaque question avant de passer à la suivante
- Bouton "Question suivante" pour progresser
- Chaque question peut avoir des options de réponse (images ou mots)

Format JSON (à implémenter) :

```
{
  "type": "lecture",
  "image": "/images/activites/2026-02-10/illustration.jpg",
  "consigneEleves": "Observe bien cette illustration",
  "questions": [
    {
      "id": "1",
      "texte": "Qui vois-tu sur cette image ?",
      "type": "observation",
      "options": [
        {
          "id": "a",
          "texte": "Un enfant",
          "image": "/images/activites/2026-02-10/q1-option-a.jpg"
        },
        {
          "id": "b",
          "texte": "Un vieil homme",
          "image": "/images/activites/2026-02-10/q1-option-b.jpg"
        }
      ],
      "feedback": "Il s'agit d'un enfant, reconnaissable à sa taille et ses vêtements."
    },
    {
      "id": "2",
      "texte": "Où se passe la scène ?",
      "type": "comprehension",
      "options": [
        { "id": "a", "texte": "Dans une forêt" },
        { "id": "b", "texte": "Dans une ville" },
        { "id": "c", "texte": "Au bord de la mer" }
      ],
      "feedback": "La scène se déroule dans une forêt, observe les arbres en arrière-plan."
    },
    {
      "id": "3",
      "texte": "Comment se sent le personnage ?",
      "type": "interpretation",
      "options": [
        { "id": "a", "texte": "Joyeux" },
        { "id": "b", "texte": "Triste" },
        { "id": "c", "texte": "Effrayé" }
      ],
      "feedback": "Son expression et sa posture montrent qu'il est joyeux."
    }
  ]
}
```


Types de questions :

1. **Observation** : "Qui/Que vois-tu ?"
2. **Localisation** : "Où se passe la scène ?"
3. **Temporalité** : "À quel moment de la journée ?"
4. **Émotion** : "Comment se sent le personnage ?"
5. **Interprétation** : "Que va-t-il se passer ensuite ?"

Adaptations multi-cycles :

- **Maternelle** : Options avec images uniquement, 2 questions simples
- **Cycle 2** : Options texte + images, 2-3 questions
- **Cycle 3** : Options texte, 3 questions dont une interprétative

Images nécessaires :

- `illustration.jpg` : L'image principale à analyser (1600-2000px)
- `q1-option-a.jpg`, `q1-option-b.jpg` : Images optionnelles pour les réponses
- `q2-option-a.jpg`, etc. : Selon les besoins

État : ~~X~~ À IMPLÉMENTER

4. Micro-défi créatif

Objectif : Proposer une consigne créative simple sans interaction numérique

Fonctionnement :

- Affichage plein écran d'une consigne créative
- Une image de référence (optionnelle) peut accompagner la consigne
- Pas d'interaction : l'activité se fait hors écran (dessin, écriture, oral)
- Durée : 5-10 minutes de création
- Retour possible en classe : exposition des productions

Format JSON (à implémenter) :

```
{
  "type": "defi",
  "consigneEleves": "Dessine un personnage à la manière de Lionel TARCHALA",
  "consigneEnseignant": "Proposez aux élèves de créer un personnage imaginaire en s'inspirant du style de l'auteur. Encouragez l'utilisation des couleurs vives et des formes géométriques.",
  "imageReference": "/images/activites/2026-03-15/exemple-style.jpg",
  "duree": 10,
  "materiel": ["Feuille A4", "Crayons de couleur", "Feutres"],
  "variantes": {
    "maternelle": "Proposez un coloriage préparé avec les formes de base.",
    "cycle2": "Demandez d'inventer un nom pour le personnage.",
    "cycle3": "Ajoutez une courte description écrite du personnage créé."
  }
}
```



```
}  
}
```

Types de défis :

1. Dessin à la manière de...

- Reproduire le style d'un illustrateur
- Créer un nouveau personnage
- Imaginer une suite de l'histoire

2. Invention de titre

- Proposer un titre pour une illustration
- Inventer un titre de livre
- Créer un slogan

3. Description créative

- Décrire une scène avec 3 mots
- Inventer une phrase d'ouverture
- Créer une courte légende

4. Défi technique

- Dessiner sans lever le crayon
- Utiliser seulement 3 couleurs
- Créer avec des formes géométriques

5. Défi narratif

- Imaginer ce qui se passe avant/après
- Inventer un dialogue entre les personnages
- Raconter du point de vue d'un personnage

Images nécessaires :

- [exemple-style.jpg](#) : Exemple de style ou d'œuvre (optionnel, 1200-1600px)
- [consigne-visuelle.jpg](#) : Support visuel pour la consigne (optionnel)

État :  À IMPLÉMENTER

État d'avancement

✓ Modules terminés (5/8)

Module 1 : Fondations ✓

Statut : 100% complet

- ☒ Configuration Vite + React 19 + Tailwind CSS 3
- ☒ Système de configuration JSON ([config.json](#), [activites.json](#))

- ☒ Hook `useCountdown` avec support J-100
- ☒ Hook `useActiviteDuJour`
- ☒ Hook `useDevMode`
- ☒ Utilitaires de dates (`dateUtils.js`)
- ☒ Structure de dossiers images créée (script automatique)
- ☒ Alias de chemins configurés (@, @components, @hooks, @utils)
- ☒ Palette de couleurs et tailles projection Tailwind

Fichiers clés :

- `src/hooks/useCountdown.js`
 - `src/hooks/useActiviteDuJour.js`
 - `src/hooks/useDevMode.js`
 - `src/utils/dateUtils.js`
 - `scripts/create-image-folders.js`
 - `vite.config.js`, `tailwind.config.js`
-

Module 2 : Page d'accueil ✓

Statut : 100% complet

- ☒ Composant `Accueil.jsx` principal
- ☒ Composant `DecompteAffichage.jsx` (badge J-??)
- ☒ Composant `BoutonActivite.jsx` (actif/inactif selon contexte)
- ☒ Composant `MessageStatut.jsx` (messages contextuels)
- ☒ Gestion des états (too-early, before, day-j, after)
- ☒ Mode d'emploi intégré
- ☒ Design responsive optimisé projection
- ☒ Animations et transitions

Comportements :

- Avant J-100 : Message "Le compte à rebours commencera à J-100", bouton désactivé
- J-100 à J-1 : Compte à rebours actif, bouton activé si activité disponible
- Jour J : Message "C'est le grand jour !", accès spécial
- Après : Message "Le salon est passé !", historique accessible

Fichiers clés :

- `src/components/Accueil/Accueil.jsx`
 - `src/components/Accueil/DecompteAffichage.jsx`
 - `src/components/Accueil/BoutonActivite.jsx`
 - `src/components/Accueil/MessageStatut.jsx`
-

Module 3 : Quiz visuel ✓

Statut : 100% complet

- ☒ Composant `QuizVisuel.jsx` (layout plein écran sans scroll)

- ☒ Composant `PropositionQuiz.jsx` (mode compact)
- ☒ Composant `AideEnseignant.jsx` (panneau aide différencié)
- ☒ Layout split 60/40 : image principale | propositions
- ☒ Support 3 propositions maximum
- ☒ Feedback immédiat avec commentaire pédagogique
- ☒ Validation visuelle (bordures vertes/rouges)
- ☒ Bouton "Réessayer" fonctionnel
- ☒ Intégration consignes enseignant/élèves
- ☒ Adaptations maternelle/cycle2/cycle3

Fonctionnalités :

- En-tête compact (py-2) avec retour, titre, auteur, aide
- Image principale 60% gauche, 3 propositions empilées 40% droite
- Clic proposition → feedback immédiat
- Commentaire pédagogique affiché
- Réessai possible via bouton dédié

Fichiers clés :

- `src/components/Activite/QuizVisuel.jsx`
 - `src/components/Activite/PropositionQuiz.jsx`
 - `src/components/Activite/AideEnseignant.jsx`
-

Module 4 : Jeu de l'intrus ✓

Statut : 100% complet

- ☒ Composant `JeuIntrus.jsx` (design optimisé plein écran)
- ☒ Composant `VignetteIntrus.jsx` (grandes vignettes)
- ☒ Layout adaptatif (3 vignettes = ligne, 4 vignettes = grille 2×2)
- ☒ Grandes vignettes occupant 95% de l'espace vertical
- ☒ Vue split-screen après sélection (50/50)
- ☒ Feedback avec badge résultat (👁️ ou X)
- ☒ Explication pédagogique affichée
- ☒ Miniatures des autres vignettes en contexte
- ☒ Bouton "Recommencer" toujours visible
- ☒ Thème violet (purple-600) distinct du quiz

Fonctionnalités :

- Vignettes énormes avant sélection (plein écran)
- Après sélection : vignette choisie 50% gauche + feedback 50% droite
- Badge résultat superposé avec animation
- Autres vignettes affichées en miniatures (contexte)
- Bouton recommencer en bas, toujours accessible

Fichiers clés :

- `src/components/Activite/JeuIntrus.jsx`

- `src/components/Activite/VignetteIntrus.jsx`
-

Module 5 : Mode développement ✓

Statut : 100% complet

- ☒ Hook `useDevMode` avec persistance `localStorage`
- ☒ Composant `DevToolbar.jsx` (bouton flottant 🗑)
- ☒ Panneau expansible avec 2 onglets (Dates, Activités)
- ☒ Navigation rapide entre les 48 dates
- ☒ Sélection directe d'une activité
- ☒ Highlight date simulée vs date réelle
- ☒ Bouton reset pour revenir à la date réelle
- ☒ Intégration dans tous les hooks (`countdown`, `activiteDuJour`)
- ☒ Persistance entre rechargements

Fonctionnalités :

- Bouton 🗑 en bas à droite (vert si actif, gris si inactif)
- Panneau 600×500px avec liste scrollable
- Clic sur date → simulation immédiate
- Clic sur activité → navigation directe
- `LocalStorage` : `devMode`, `devSimulatedDate`

Fichiers clés :

- `src/hooks/useDevMode.js`
 - `src/components/Dev/DevToolbar.jsx`
-

🕒 Modules en cours / à faire (3/8)

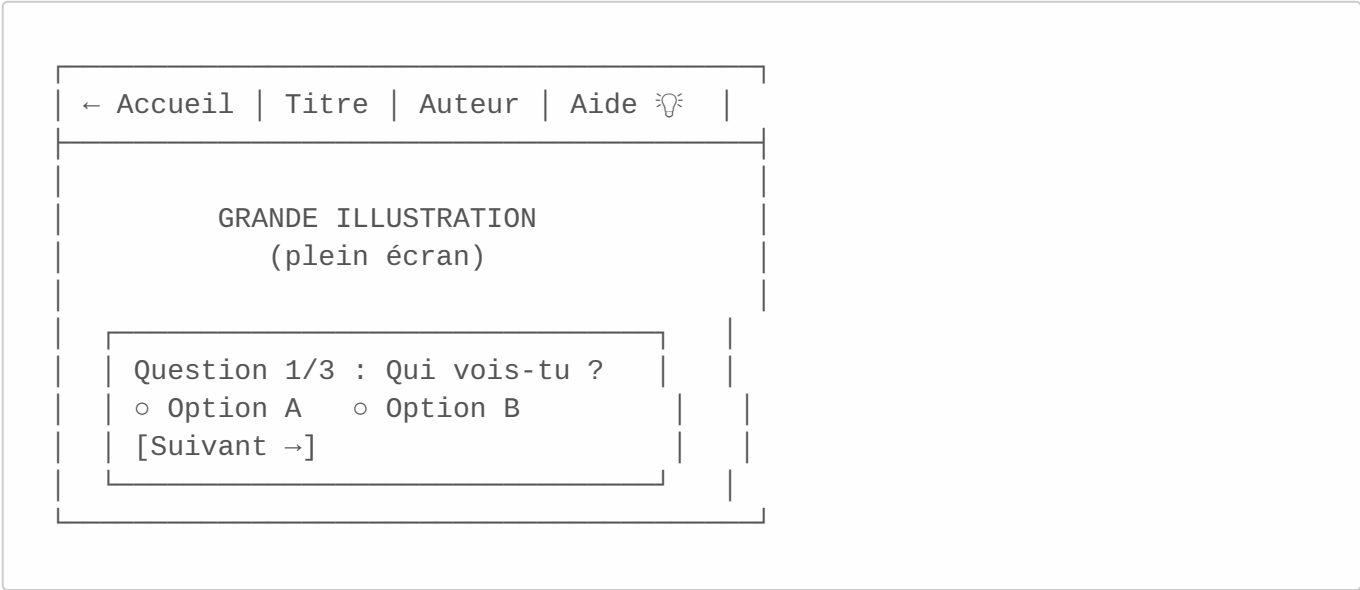
Module 6 : "Je lis une image" 🕒

Statut : 0% - À implémenter

À faire :

- ☐ Créer composant `LectureImage.jsx`
- ☐ Gérer état `currentQuestion` (`useState`)
- ☐ Implémenter navigation entre questions
- ☐ Créer composant `QuestionLecture.jsx`
- ☐ Support options avec images ou texte
- ☐ Affichage feedback par question
- ☐ Bouton "Question suivante"
- ☐ Bouton "Terminer" sur dernière question
- ☐ Intégration dans `App.jsx` (type 'lecture')
- ☐ Adaptation multi-cycles (2-3 questions selon niveau)

Complexité estimée : Moyenne
Temps estimé : 4-6 heures
Priorité : Haute (type d'activité essentiel)
Design cible :



Module 7 : Micro-défi créatif 🧩

Statut : 0% - À implémenter

À faire :

- ☐ Créer composant `DefiCreatif.jsx`
- ☐ Affichage consigne plein écran
- ☐ Support image de référence optionnelle
- ☐ Affichage durée estimée
- ☐ Affichage matériel nécessaire
- ☐ Pas d'interaction (projection simple)
- ☐ Design épuré et lisible
- ☐ Intégration dans `App.jsx` (type 'defi')
- ☐ Variantes par cycle affichées dans aide

Complexité estimée : Faible
Temps estimé : 2-3 heures
Priorité : Moyenne (le plus simple des 4 types)

Design cible :



de Lionel TARCHALA

[Image de référence optionnelle]

🕒 Durée : 10 minutes

📦 Matériel : Feuille, crayons

Module 8 : Intégration finale et polish ⌚

Statut : 50% - En cours

✔ Déjà fait :

- ☒ Router de base dans App.jsx (state-based)
- ☒ Navigation retour accueil
- ☒ Chargement dynamique quiz et intrus
- ☒ Gestion états loading/error
- ☒ PropTypes sur tous les composants
- ☒ Design plein écran sans scroll

⌚ À faire :

- ☐ Intégrer type 'lecture' dans App.jsx
- ☐ Intégrer type 'defi' dans App.jsx
- ☐ Transitions animées entre vues (Framer Motion ?)
- ☐ Écran de chargement amélioré
- ☐ Gestion d'erreur robuste (images manquantes, JSON invalide)
- ☐ Tests finaux tous types d'activités
- ☐ Optimisation performances (lazy loading composants)
- ☐ Validation accessibilité complète (WCAG AA)
- ☐ Tests sur vidéoprojecteur réel
- ☐ Ajustements tailles police si nécessaire
- ☐ Documentation utilisateur enseignants

Complexité estimée : Moyenne

Temps estimé : 6-8 heures

Priorité : Haute (finalisation avant prod)

📈 Progression globale

 62.5% (5/8 modules)

Détail par module :

- | | | |
|-----------------------------|--|------|
| ✔ Module 1 : Fondations |  | 100% |
| ✔ Module 2 : Page d'accueil |  | 100% |
| ✔ Module 3 : Quiz visuel |  | 100% |

✓	Module 4 : Jeu de l'intrus	[████████████████████]	100%
✓	Module 5 : Mode développement	[████████████████████]	100%
🕒	Module 6 : "Je lis une image"	[░░░░░░░░░░░░░░░░]	0%
🕒	Module 7 : Micro-défi créatif	[░░░░░░░░░░░░░░░░]	0%
🕒	Module 8 : Intégration finale	[██████████░░░░░░░░]	50%

Estimation pour finalisation :

- Module 6 : 4-6 heures
- Module 7 : 2-3 heures
- Module 8 : 6-8 heures
- **Total : 12-17 heures de développement**

🎯 Prochaines étapes immédiates

Sprint 1 : Types d'activités (10h)

1. Implémenter "Je lis une image" (6h)
 - Composant LectureImage
 - Système questions/réponses
 - Navigation entre questions
 - Tests avec données JSON
2. Implémenter "Micro-défi créatif" (4h)
 - Composant DefiCreatif
 - Affichage consigne + image
 - Tests avec données JSON

Sprint 2 : Contenu (20h) 3. Rédiger les 48 activités (15h)

- 12 quiz visuels
 - 12 jeux de l'intrus
 - 12 lectures d'images
 - 12 défis créatifs
4. Collecter/créer les images (5h)
- 13 photos auteurs
 - Images pour les 48 activités

Sprint 3 : Finalisation (8h) 5. Intégration et tests (4h)

- Intégrer types lecture et defi dans App.jsx
 - Tests sur vidéoprojecteur
 - Ajustements design
- ## 6. Polish et déploiement (4h)
- Animations transitions
 - Optimisations performances
 - Déploiement sur micetf.fr

Architecture technique

Stack technologique

```
Frontend
├─ React 19.2.0           # Library UI
├─ Vite 7.2.4             # Build tool & dev server
├─ Tailwind CSS 3.4.1     # Framework CSS utility-first
├─ PropTypes              # Validation des props

Development Tools
├─ ESLint                 # Linter JavaScript
├─ pnpm                   # Gestionnaire de packages
├─ Node.js 16+           # Runtime JavaScript
```

Principes architecturaux

1. **Composants fonctionnels React** : Utilisation exclusive de hooks
2. **Design responsive** : Optimisé pour projection (1920×1080)
3. **Pas de backend** : Application 100% statique (fichiers JSON)
4. **Pas de base de données** : Données stockées dans [/public/data/](#)
5. **Mode développement intégré** : Simulation de dates pour tests
6. **Accessibility first** : ARIA labels, navigation clavier

Prérequis

Logiciels requis

```
Node.js >= 16.0.0
pnpm >= 8.0.0   (ou npm >= 9.0.0)
Git >= 2.30.0
```

Vérification des versions

```
node --version    # v16.x.x ou supérieur
pnpm --version    # 8.x.x ou supérieur
git --version     # 2.30.x ou supérieur
```

Installation de pnpm (si nécessaire)

```
npm install -g pnpm
```

Installation

1. Cloner le dépôt

```
git clone https://github.com/micetf/un-auteur-dans-sa-classe.git
cd un-auteur-dans-sa-classe
```

2. Installer les dépendances

```
pnpm install
```

3. Créer la structure des dossiers d'images

```
node scripts/create-image-folders.js
```

Cette commande crée automatiquement :

- `public/images/auteurs/` (13 photos attendues)
- `public/images/activites/2026-XX-XX/` (48 dossiers pour les activités)

4. Lancer le serveur de développement

```
pnpm dev
```

L'application sera accessible sur **`http://localhost:3000`**

5. Build de production

```
pnpm build
```

Les fichiers optimisés seront générés dans `/dist/`

Configuration

Fichiers de configuration principaux

`public/data/config.json`

Configuration globale de l'application.


```

{
  "titre": "Un auteur dans sa classe",
  "dateSalon": "2026-04-25",
  "dateDebutCompteAREbours": 100,
  "description": "Une activité quotidienne de 10 minutes pour découvrir les auteurs et illustrateurs invités au salon du livre.",
  "messageAvantDebut": "Le compte à rebours commencera à J-100, soit le 15 janvier 2026. Revenez à partir de cette date !",
  "messageApres": "Le salon est passé ! Merci pour votre participation. Rendez-vous l'année prochaine !",
  "messageJourJ": "C'est le grand jour ! Bon salon du livre !",
  "auteurs": [
    {
      "id": "tarchala",
      "nom": "Lionel TARCHALA",
      "type": "illustrateur",
      "voteCE2": true,
      "bio": "Illustrateur passionné par les univers fantastiques...",
      "photo": "/images/auteurs/tarchala.jpg"
    }
    // ... 12 autres auteurs
  ]
}

```

Paramètres clés :

- **dateSalon** : Date du salon (format ISO YYYY-MM-DD)
- **dateDebutCompteAREbours** : Nombre de jours avant le salon pour commencer (100 = J-100)
- **auteurs** : Liste des 13 auteurs/illustrateurs invités

public/data/activites.json

Contenu des 48 activités (structure complète dans la section "Gestion des activités").

📁 Structure du projet

```

un-auteur-dans-sa-classe/
├── public/                                # Assets statiques
│   ├── data/
│   │   ├── config.json                  # Configuration globale
│   │   └── activites.json               # Contenu des 48 activités
│   └── images/
│       ├── auteurs/                     # Photos des 13 auteurs (400×400px)
│       │   ├── tarchala.jpg
│       │   ├── broncard.jpg
│       │   └── ...
│       └── activites/                   # Images des 48 activités
│           ├── 2026-01-15/              # Quiz visuel
│           └── image-principale.jpg

```



```

├── proposition-a.jpg
├── proposition-b.jpg
├── proposition-c.jpg
├── 2026-01-16/      # Jeu de l'intrus
│   ├── vignette-1.jpg
│   ├── vignette-2.jpg
│   ├── vignette-3.jpg
│   └── vignette-4.jpg
├── 2026-01-19/      # Lecture d'image
│   ├── illustration.jpg
│   ├── q1-option-a.jpg
│   └── q2-option-a.jpg
├── 2026-01-20/      # Défi créatif
│   └── exemple-style.jpg
└── ...

scripts/
├── create-image-folders.js # Génère la structure des dossiers
└── verify-images.js       # Vérifie la présence des images

src/
├── components/
│   ├── Accueil/          # Page d'accueil
│   │   ├── Accueil.jsx
│   │   ├── DecompteAffichage.jsx
│   │   ├── BoutonActivite.jsx
│   │   ├── MessageStatut.jsx
│   │   └── index.js
│   ├── Activite/         # Types d'activités
│   │   ├── QuizVisuel.jsx ✓
│   │   ├── PropositionQuiz.jsx ✓
│   │   ├── JeuIntrus.jsx ✓
│   │   ├── VignetteIntrus.jsx ✓
│   │   ├── LectureImage.jsx ⌚
│   │   ├── QuestionLecture.jsx ⌚
│   │   ├── DefiCreatif.jsx ⌚
│   │   ├── AideEnseignant.jsx ✓
│   │   └── index.js
│   └── Dev/              # Outils de développement
│       ├── DevToolbar.jsx
│       └── index.js
├── hooks/
│   ├── useCountdown.js   # Gestion du compte à rebours
│   ├── useActiviteDuJour.js
│   └── useDevMode.js     # Mode développement
├── utils/
│   └── dateUtils.js       # Utilitaires de dates
├── App.jsx               # Composant racine
├── main.jsx              # Point d'entrée React
└── index.css              # Styles globaux + Tailwind

```


— index.html	# Template HTML
— package.json	# Dépendances npm
— vite.config.js	# Configuration Vite
— tailwind.config.js	# Configuration Tailwind
— postcss.config.js	# Configuration PostCSS
— eslint.config.js	# Configuration ESLint
— README.md	# Ce fichier

Légende :

- ✓ Fichier implémenté et fonctionnel
- ⌚ Fichier à créer

↩ Guide de développement

Conventions de code

Composants React

```
/**
 * Documentation JSDoc complète en français
 * @param {Object} props - Description
 * @returns {JSX.Element} Description
 */
function MonComposant({ prop1, prop2 }) {
  // Code...
  return <div>...</div>;
}

MonComposant.propTypes = {
  prop1: PropTypes.string.isRequired,
  prop2: PropTypes.number,
};

export default MonComposant;
```

Classes Tailwind

```
// ✓ BON : Classes organisées et lisibles
<div className="
  flex items-center justify-between
  bg-white rounded-xl shadow-lg
  p-6 mb-4
  transition-all duration-300
">

// ✗ MAUVAIS : Tout sur une ligne
```



```
<div className="flex items-center justify-between bg-white rounded-xl shadow-lg p-6 mb-4 transition-all duration-300">
```

Hooks personnalisés

useCountdown

Gère le compte à rebours jusqu'à la date du salon.

```
const countdown = useCountdown(
  "2026-04-25", // Date cible
  100, // Jours avant début (J-100)
  simulatedDate // Date simulée (mode dev)
);

// Retourne :
// {
//   daysRemaining: number,
//   status: 'too-early' | 'before' | 'day-j' | 'after'
// }
```

useActiviteDuJour

Vérifie si une activité est disponible aujourd'hui.

```
const { hasActivity, isLoading } = useActiviteDuJour(
  datesActivites,
  simulatedDate
);
```

useDevMode

Gère le mode développement avec simulation de dates.

```
const devMode = useDevMode();

// Méthodes disponibles :
devMode.isDevMode; // boolean
devMode.simulatedDate; // string | null
devMode.toggleDevMode(); // Activer/désactiver
devMode.setDate(date); // Simuler une date
devMode.resetDate(); // Revenir à la date réelle
devMode.getCurrentDate(); // Obtenir la date courante
```



Format JSON complet

public/data/activites.json

```
{
  "activites": [
    {
      "date": "2026-01-15",
      "type": "quiz",
      "titre": "Bienvenue dans l'univers de Lionel TARCHALA",
      "auteurId": "tarchala",
      "consigneEnseignant": "Première activité pour découvrir le travail de Lionel TARCHALA. Laissez les élèves observer l'image puis discuter collectivement.",
      "consigneEleves": "Observez bien cette illustration de Lionel TARCHALA. Quel animal voyez-vous ?",
      "image": "/images/activites/2026-01-15/image-principale.jpg",
      "propositions": [
        {
          "id": "a",
          "texte": "Un dragon",
          "image": "/images/activites/2026-01-15/proposition-a.jpg",
          "correct": true,
          "commentaire": "Bravo ! C'est bien un dragon. Lionel TARCHALA adore dessiner des créatures fantastiques avec des couleurs vives."
        },
        {
          "id": "b",
          "texte": "Un chat",
          "correct": false,
          "commentaire": "Non, ce n'est pas un chat. Regardez bien les écailles et les ailes !"
        },
        {
          "id": "c",
          "texte": "Un oiseau",
          "correct": false,
          "commentaire": "Les ailes peuvent faire penser à un oiseau, mais observez la queue et le corps."
        }
      ],
      "adaptations": {
        "maternelle": "Privilégiez le pointage sur l'image. Faites nommer les couleurs et les formes.",
        "cycle2": "Les élèves peuvent lire les propositions. Demandez-leur de justifier leur réponse.",
        "cycle3": "Analysez les techniques d'illustration utilisées par l'artiste (couleurs, traits, composition)."
      }
    },
    {
      "date": "2026-01-16",
```



```

    "type": "intrus",
    "titre": "Quel est l'intrus ?",
    "auteurId": "broncard",
    "consigneEnseignant": "Laissez les élèves observer les quatre
images puis demandez-leur d'identifier l'intrus en justifiant leur choix.",
    "consigneEleves": "Laquelle de ces images est l'intrus ?",
    "vignettes": [
        {
            "id": "1",
            "image": "/images/activites/2026-01-16/vignette-1.jpg",
            "titre": "Image 1",
            "isIntrus": false,
            "explication": "Cette image partage le même style
fantastique que les autres."
        },
        {
            "id": "2",
            "image": "/images/activites/2026-01-16/vignette-2.jpg",
            "titre": "Image 2",
            "isIntrus": true,
            "explication": "Bravo ! C'est l'intrus car c'est la
seule illustration en noir et blanc. Toutes les autres utilisent des
couleurs vives."
        },
        {
            "id": "3",
            "image": "/images/activites/2026-01-16/vignette-3.jpg",
            "titre": "Image 3",
            "isIntrus": false,
            "explication": "Cette image utilise les mêmes couleurs
pastel que les autres."
        },
        {
            "id": "4",
            "image": "/images/activites/2026-01-16/vignette-4.jpg",
            "titre": "Image 4",
            "isIntrus": false,
            "explication": "Le style poétique de cette image
correspond aux autres."
        }
    ],
    "adaptations": {
        "maternelle": "Laissez les élèves pointer l'intrus et
expliquer oralement pourquoi.",
        "cycle2": "Demandez d'identifier précisément la différence
(couleur, style, sujet).",
        "cycle3": "Analysez les techniques artistiques : aquarelle,
crayon, numérique, etc."
    }
},
{
    "date": "2026-01-19",
    "type": "lecture",
    "titre": "Je lis une image",
    "auteurId": "vaguez",

```



```

"consigneEnseignant": "Guidez les élèves à travers les trois
questions pour analyser progressivement l'illustration.",
"consigneEleves": "Observe bien cette illustration de
Clémentine VAQUEZ",
"image": "/images/activites/2026-01-19/illustration.jpg",
"questions": [
  {
    "id": "1",
    "texte": "Qui vois-tu sur cette image ?",
    "type": "observation",
    "options": [
      {
        "id": "a",
        "texte": "Un enfant",
        "image": "/images/activites/2026-01-19/q1-
option-a.jpg"
      },
      {
        "id": "b",
        "texte": "Un vieil homme",
        "image": "/images/activites/2026-01-19/q1-
option-b.jpg"
      }
    ],
    "feedback": "Il s'agit d'un enfant, reconnaissable à sa
petite taille et ses vêtements colorés."
  },
  {
    "id": "2",
    "texte": "Où se passe la scène ?",
    "type": "comprehension",
    "options": [
      { "id": "a", "texte": "Dans une forêt" },
      { "id": "b", "texte": "Dans une ville" },
      { "id": "c", "texte": "Au bord de la mer" }
    ],
    "feedback": "La scène se déroule dans une forêt.
Observe les grands arbres et la végétation autour du personnage."
  },
  {
    "id": "3",
    "texte": "Comment se sent le personnage ?",
    "type": "interpretation",
    "options": [
      { "id": "a", "texte": "Joyeux" },
      { "id": "b", "texte": "Triste" },
      { "id": "c", "texte": "Effrayé" }
    ],
    "feedback": "Le personnage semble joyeux. Son sourire
et sa posture dynamique le montrent clairement."
  }
],
"adaptations": {
  "maternelle": "Utilisez uniquement les deux premières
questions avec des options illustrées.",

```



```

        "cycle2": "Proposez les trois questions en guidant la
lecture des options textuelles.",
        "cycle3": "Ajoutez une question bonus : 'Que va-t-il se
passer ensuite selon toi ?'"
    }
},
{
    "date": "2026-01-20",
    "type": "defi",
    "titre": "Dessine à la manière de...",
    "auteurId": "jalbert",
    "consigneEnseignant": "Proposez aux élèves de créer un
personnage en s'inspirant du style de Philippe JALBERT. Encouragez
l'utilisation de formes simples et de couleurs vives.",
    "consigneEleves": "Dessine un animal imaginaire à la manière de
Philippe JALBERT",
    "imageReference": "/images/activites/2026-01-20/exemple-
style.jpg",
    "duree": 10,
    "materiel": ["Feuille A4", "Crayons de couleur", "Feutres"],
    "adaptations": {
        "maternelle": "Proposez un coloriage préparé avec les
formes de base de l'artiste.",
        "cycle2": "Demandez d'inventer un nom rigolo pour l'animal
créé.",
        "cycle3": "Ajoutez une courte description écrite de
l'animal imaginaire (3-4 phrases)."
    }
}
// ... 44 autres activités
],
"datesActivites": [
    "2026-01-15",
    "2026-01-16",
    "2026-01-19",
    "2026-01-20",
    "2026-01-22",
    "2026-01-23",
    "2026-01-26",
    "2026-01-27",
    "2026-01-29",
    "2026-01-30",
    "2026-02-02",
    "2026-02-03",
    "2026-02-05",
    "2026-02-06",
    "2026-02-23",
    "2026-02-24",
    "2026-02-26",
    "2026-02-27",
    "2026-03-02",
    "2026-03-03",
    "2026-03-05",
    "2026-03-06",
    "2026-03-09",

```



```
"2026-03-10",
"2026-03-12",
"2026-03-13",
"2026-03-16",
"2026-03-17",
"2026-03-19",
"2026-03-20",
"2026-03-23",
"2026-03-24",
"2026-03-26",
"2026-03-27",
"2026-03-30",
"2026-03-31",
"2026-04-02",
"2026-04-03",
"2026-04-06",
"2026-04-07",
"2026-04-09",
"2026-04-10",
"2026-04-13",
"2026-04-14",
"2026-04-16",
"2026-04-17",
"2026-04-20",
"2026-04-21",
"2026-04-23",
"2026-04-24"
]
}
```

🔧 Mode développement

Activation

Cliquer sur le bouton 🛠 en bas à droite de l'écran.

Fonctionnalités

1. Panneau de contrôle

- **Dates** : Liste des 48 dates avec activités
- **Activités** : Liste des 48 activités avec détails
- **Sélection** : Clic pour simuler une date/activité
- **Reset** : Bouton pour revenir à la date réelle

2. Persistance

Les paramètres du mode dev sont sauvegardés dans `localStorage` :

- `devMode` : État activé/désactivé
- `devSimulatedDate` : Date actuellement simulée

3. Navigation rapide

```
// En tant que développeur, vous pouvez forcer une date en console :
localStorage.setItem("devMode", "true");
localStorage.setItem("devSimulatedDate", "2026-02-15");
window.location.reload();
```

Déploiement

URL de production

<https://micetf.fr/un-auteur-dans-sa-classe>

Build de production

```
pnpm build
```

Génère les fichiers optimisés dans `/dist/` :

- HTML minifié
- CSS avec Tailwind purgé (uniquement les classes utilisées)
- JavaScript avec code splitting
- Assets optimisés

Configuration Vite pour sous-dossier

vite.config.js

```
import { defineConfig } from "vite";
import react from "@vitejs/plugin-react";
import { resolve } from "path";

export default defineConfig({
  plugins: [react()],
  base: "/un-auteur-dans-sa-classe/", // ← Important pour micetf.fr
  resolve: {
    alias: {
      "@": resolve(__dirname, "./src"),
      "@components": resolve(__dirname, "./src/components"),
      "@hooks": resolve(__dirname, "./src/hooks"),
      "@utils": resolve(__dirname, "./src/utils"),
    },
  },
  build: {
    outDir: "dist",
    assetsDir: "assets",
  },
});
```



```
    },  
  });
```

Déploiement sur serveur Apache

1. Build

```
pnpm build
```

2. Upload via FTP/SSH

```
# Via rsync  
rsync -avz dist/ user@micetf.fr:/var/www/micetf.fr/un-auteur-dans-sa-classe/  
  
# Via SCP  
scp -r dist/* user@micetf.fr:/var/www/micetf.fr/un-auteur-dans-sa-classe/
```

3. Configuration Apache

Créer `.htaccess` dans `/un-auteur-dans-sa-classe/` :

```
<IfModule mod_rewrite.c>  
  RewriteEngine On  
  RewriteBase /un-auteur-dans-sa-classe/  
  
  # Rediriger toutes les requêtes vers index.html (SPA)  
  RewriteRule ^index\.html$ - [L]  
  RewriteCond %{REQUEST_FILENAME} !-f  
  RewriteCond %{REQUEST_FILENAME} !-d  
  RewriteRule . /un-auteur-dans-sa-classe/index.html [L]  
  
  # Cache des assets  
  <FilesMatch "\.(jpg|jpeg|png|gif|svg|webp|css|js)$">  
    Header set Cache-Control "max-age=31536000, public"  
  </FilesMatch>  
  
  # Compression GZIP  
  <IfModule mod_deflate.c>  
    AddOutputFilterByType DEFLATE text/html text/plain text/xml text/css  
    text/javascript application/javascript  
  </IfModule>  
</IfModule>
```

Vérification post-déploiement


```
# Tester localement la version de production
pnpm preview

# Vérifier les chemins d'assets
curl -I https://micetf.fr/un-auteur-dans-sa-classe/assets/index-xxx.js
curl -I https://micetf.fr/un-auteur-dans-sa-classe/images/auteurs/tarchala.jpg

# Vérifier la compression
curl -H "Accept-Encoding: gzip" -I https://micetf.fr/un-auteur-dans-sa-classe/
```

Checklist de déploiement

- ☐ Build réussie sans erreurs (`pnpm build`)
- ☐ Toutes les images présentes dans `/public/images/`
- ☐ Fichiers JSON validés (syntax JSON valide)
- ☐ `.htaccess` configuré correctement
- ☐ Test en local avec `pnpm preview`
- ☐ Upload des fichiers sur le serveur
- ☐ Vérification URL production accessible
- ☐ Test navigation (accueil → activité → retour)
- ☐ Test mode dev désactivé en production
- ☐ Test responsive sur plusieurs résolutions
- ☐ Vérification console navigateur (pas d'erreur)

Contribution

Workflow Git

1. Créer une branche

```
git checkout -b feature/lecture-image
# ou
git checkout -b fix/bug-affichage-quiz
```

2. Développer et commiter

```
git add .
git commit -m "feat: ajout du type d'activité 'lecture d'image'"
```

Convention de commit :

- `feat`: Nouvelle fonctionnalité

- **fix**: Correction de bug
- **docs**: Documentation
- **style**: Formatage (pas de changement de code)
- **refactor**: Refactoring
- **test**: Ajout de tests
- **chore**: Maintenance

3. Pousser et créer une Pull Request

```
git push origin feature/lecture-image
```

Checklist avant PR

- ☐ Le code compile sans erreur (**pnpm build**)
- ☐ Aucun warning ESLint
- ☐ PropTypes définis sur tous les composants
- ☐ Documentation JSDoc à jour
- ☐ Mode dev testé
- ☐ Responsive vérifié (1920×1080 minimum)
- ☐ Accessibilité vérifiée (ARIA labels)

Troubleshooting

Problème : Les images ne s'affichent pas

Cause possible : Chemin incorrect dans le JSON

✓ **Solution** :

```
// ✓ BON
"image": "/images/activites/2026-01-15/image-principale.jpg"

// ✗ MAUVAIS
"image": "images/activites/2026-01-15/image-principale.jpg" // Manque le /
```

Problème : Le compte à rebours ne démarre pas

Symptôme : Message "Le compte à rebours n'a pas encore commencé"

✓ **Solution** :

1. Vérifier **dateDebutCompteAREbours**: 100 dans **config.json**
2. Utiliser le mode dev pour simuler une date après le 15 janvier 2026

Problème : Build échoue

✓ **Solutions courantes** :


```
# Réinstaller les dépendances
rm -rf node_modules pnpm-lock.yaml
pnpm install

# Vérifier Tailwind
pnpm remove tailwindcss
pnpm add -D tailwindcss@^3.4.1 postcss autoprefixer
```

Statistiques du projet

État actuel (Décembre 2024) :

- ✓ 5/8 modules terminés (62.5%)
- ✓ 2/4 types d'activités implémentés
- ⌚ 2/4 types à implémenter
- ⌚ 12-17h de développement restantes

Lignes de code :

- React : ~2800 lignes
- CSS/Tailwind : ~200 lignes
- JSON : ~1500 lignes (à compléter)

Licence

Ce projet est la propriété de **MiCetF** (Frédéric MISERY).

Tous droits réservés © 2024-2026 MiCetF

Contact et support

Mainteneur : Frédéric MISERY

Site : <https://micetf.fr>

Email : webmaster@micetf.fr

Projet : <https://micetf.fr/un-auteur-dans-sa-classe>

Roadmap

Version 1.0 (Production - Mars 2026)

- ☒ Fondations et architecture
- ☒ Page d'accueil avec compte à rebours J-100
- ☒ Quiz visuel (3 propositions)
- ☒ Jeu de l'intrus (3-4 vignettes)
- ☒ Mode développement
- ☐ "Je lis une image" (2-3 questions)

- ☐ Micro-défi créatif
- ☐ 48 activités complètes avec images
- ☐ Tests finaux et déploiement

Version 1.1 (Améliorations - Avril 2026)

- ☐ Statistiques d'utilisation (anonymes)
- ☐ Export PDF des activités
- ☐ Mode impression enseignant
- ☐ Galerie des productions d'élèves (optionnel)

Version 2.0 (Année suivante)

- ☐ Mode hors-ligne (PWA)
- ☐ Son/narration optionnelle
- ☐ Interface d'administration pour éditer les activités
- ☐ Personnalisation des couleurs par école
- ☐ Multi-langue (occitan, anglais)

Dernière mise à jour : 05 janvier 2026

Version du document : 2.0

Auteur : MiCetF (Frédéric MISERY)