

CINEchrono TRAVEL 開発引き継ぎドキュメント

プロジェクト概要

プロジェクト名: CINEchrono TRAVEL

ドメイン: cinechrono.com (取得済み)

目的: 歴史的瞬間と映画・漫画・ゲームを年表で繋ぐWebアプリ

ターゲット: 中高生（世界史学習者）

完了した作業

1. 開発環境構築

- ✓ Node.js (v20.x) インストール済み
- ✓ Git インストール済み
- ✓ VSCode 使用中

2. プロジェクト作成

- ✓ Reactプロジェクト作成: `npx create-react-app cinechrono`
- ✓ パス: `/Users/hiroec/Desktop/cinechrono`
- ✓ Firebase インストール済み: `npm install firebase`
- ✓ Tailwind CSS v3.4.1 インストール済み

3. コード実装

- ✓ `src/App.js`: メインアプリケーションコード配置済み
- ✓ `src/index.css`: Tailwind CSS 設定済み
- ✓ `tailwind.config.js`: 設定完了
- ✓ ローカル起動確認済み: `npm start` → <http://localhost:3000>

4. GitHub 連携

- ✓ GitHubリポジトリ作成完了
 - ✓ URL: <https://github.com/chiroe203/cinechrono>
 - ✓ 初回プッシュ完了
-


🔥 Firebase 設定情報

プロジェクト情報

javascript

```
const firebaseConfig = {  
  apiKey: "AlzaSyDyJ6lhfEAf6Bsf7LTQ5YByG8T-ou6cXwE",  
  authDomain: "cinechrono-1c1a8.firebaseio.com",  
  projectId: "cinechrono-1c1a8",  
  storageBucket: "cinechrono-1c1a8.firebaseio.com",  
  messagingSenderId: "1029924381560",  
  appId: "1:1029924381560:web:5c36f1b9ac2ed2f7a09e8d"  
};
```





Firestore Database

- URL: <https://console.firebase.google.com/project/cinechrono-1c1a8>
- ☒ Firestore Database 作成済み
-  Authentication 未設定（次のステップで設定）

🎨 アプリの主要機能

ユーザー向け機能

1. 年表表示

- 3階層構造: 大時代（古代・中世など） → 中時代（ローマ帝国など） → 年号
- 色分け:  歴史イベント /  映画 /  漫画 /  ゲーム
- 時計アイコン付き中時代表示
- 点線で年表に接続

2. 詳細モールド

- あらすじ表示
- アフィリエイトリンク（Amazon Prime、Netflix等）
- トピック記事リンク（Note記事へ）

3. 3ページ構成

- 年表と物語（メイン）
- CINEchrono TRAVELとは（About）



- 記事一覧（Note埋め込み）

管理者機能

1. 管理モード切り替え

- フッター小歯車 → パスワード入力 → ピンク歯車表示
- パスワード: `cinechrono2024`

2. 管理画面（2タブ）

-  作品追加: カテゴリ/時代/年代/タイトル/あらすじ/リンク/トピック記事
-  イベント追加: 時代/年代/タイトル/概要/詳細/トピック記事
- 登録済みコンテンツ一覧 + 削除機能

ファイル構成

```
cinechrono/  
├── public/  
│   ├── index.html      # タイトル、メタタグ  
│   └── favicon.ico     # ファビコン（未設定）  
├── src/  
│   ├── App.js          # メインコード（Claudeで作成済み）  
│   ├── firebase.js     # Firebase設定（次で作成）  
│   ├── index.js        # React起動ファイル  
│   └── index.css        # Tailwind CSS  
├── node_modules/       # ライブラリ（Git管理外）  
├── package.json         # 依存関係  
├── tailwind.config.js   # Tailwind設定  
└── .gitignore           # Git無視ファイル
```

次のステップ（残作業）

B: Firebase データベース連携（重要！）

1. `src/firebase.js` 作成

```
javascript
```

```
import { initializeApp } from 'firebase/app';
import { getFirestore, collection, addDoc, getDocs, deleteDoc, doc } from 'firebase/firestore';
import { getAuth, signInWithEmailAndPassword } from 'firebase/auth';

const firebaseConfig = {
  apiKey: "AlzaSyDyJ6lhfEaf6Bsf7LTQ5YByG8T-ou6cXwE",
  authDomain: "cinechrono-1c1a8.firebaseio.com",
  projectId: "cinechrono-1c1a8",
  storageBucket: "cinechrono-1c1a8.firebaseio.com",
  messagingSenderId: "1029924381560",
  appId: "1:1029924381560:web:5c36f1b9ac2ed2f7a09e8d"
};

const app = initializeApp(firebaseConfig);
export const db = getFirestore(app);
export const auth = getAuth(app);
```

2. Firebase Console での設定

1. Firestore セキュリティルール

```
javascript

rules_version = '2';
service cloud.firestore {
  match /databases/{database}/documents {
    match /timeline/{document=**} {
      allow read: if true;
      allow write: if request.auth != null;
    }
  }
}
```

2. Authentication 有効化

- 左メニュー → Authentication → 始める
- メール/パスワード を有効化
- Users タブ → ユーザー追加
- メール: hiroec@example.com (任意)
- パスワード: 強力なパスワード

3. App.js にFirebase機能を統合

- データ読み込み: `getDocs(collection(db, 'timeline'))`

- データ追加: `addDoc(collection(db, 'timeline'), data)`
- データ削除: `deleteDoc(doc(db, 'timeline', id))`

注意: この統合は複雑なので、Claudeに「FirebaseとApp.jsを統合したい」と依頼すること

C: Vercel デプロイ

1. Vercel アカウント作成

- <https://vercel.com>
- GitHubアカウントでログイン

2. プロジェクト接続

1. 「New Project」
2. GitHub リポジトリ `chiroe203/cinechrono` を選択
3. Framework Preset: **Create React App**
4. Environment Variables (環境変数) :

```
REACT_APP_FIREBASE_API_KEY=AlzaSyDyJ6lhfEAf6Bsf7LTQ5YByG8T-ou6cXwE
REACT_APP_FIREBASE_AUTH_DOMAIN=cinechrono-1c1a8.firebaseio.com
REACT_APP_FIREBASE_PROJECT_ID=cinechrono-1c1a8
```

5. 「Deploy」 クリック

3. カスタムドメイン設定

1. Vercel → Settings → Domains
 2. `cinechrono.com` 入力
 3. DNS設定 (ドメイン管理画面で) :
 - **Aレコード:** `76.76.21.21`
 - **CNAME (www):** `cname.vercel-dns.com`
 4. 最大48時間待機 (通常は数時間)
-

今後の更新フロー

コード更新時

```
bash

# 1. ファイルを編集
# 2. ローカルで確認
npm start

# 3. Gitにコミット
git add .
git commit -m "機能追加: 検索機能実装"
git push

# 4. Vercelが自動デプロイ (1-2分)
```

Claudeとの連携

1. 「App.jsのXX機能を修正したい」と依頼
 2. Claudeが修正コードを提示
 3. 該当箇所を修正
 4. 上記フローでデプロイ
-

収益化戦略

アフィリエイト登録（必須）

1. もしもアフィリエイト: <https://af.moshimo.com>
 - Amazon・楽天の商品リンク
2. A8.net: <https://www.a8.net>
 - U-NEXT、Hulu等の動画配信
3. バリューコマー스: <https://www.valuecommerce.ne.jp>
 - Yahoo!ショッピング

コンテンツ充実

- 目標: 各時代 5-10作品
- 合計: 50-100作品

- 注力時代: 第二次世界大戦、幕末、ローマ帝国
-

🔧 トラブルシューティング

よくあるエラー

1. `npm start` でエラー

```
bash

# node_modules削除 → 再インストール
rm -rf node_modules package-lock.json
npm install
npm start
```

2. Gitプッシュでエラー

```
bash

# Personal Access Token が必要
# GitHub → Settings → Developer settings → Tokens
```

3. Vercelデプロイエラー

- Build Command: `npm run build`
 - Output Directory: `build`
 - Install Command: `npm install`
-

📞 次のチャットで伝えること

「CINEchrono TRAVELの開発を続けたい。引き継ぎドキュメントがある。現在、GitHubアップロード完了。次はFirebaseデータベース連携（ステップB）から進めたい。」

と伝えれば、Claudeがスムーズに対応します！

📝 重要なメモ

- ローカルパス: `/Users/hiroec/Desktop/cinechrono`
- GitHub: <https://github.com/chiroe203/cinechrono>
- 管理画面パスワード: `cinechrono2024`（App.jsの `ADMIN_PASSWORD` で変更可能）

- **Firebase Project ID:** `cinechrono-1c1a8`
-

✨ 完成イメージ

ユーザー



cinechrono.com にアクセス



年表を閲覧



作品クリック → アフィリエイトリンク



購入 → 収益発生

管理者（あなた）



フッター歯車 → パスワード入力



管理画面 → 作品追加



Firebase に保存



全ユーザーに即反映

🧑🏻‍💻 現在の進捗: 60%

- ☒ プロジェクト作成
- ☒ コード実装
- ☒ GitHub 連携
- ⌚ Firebase データベース連携（次）
- ⌚ Vercel デプロイ
- ⌚ ドメイン接続

頑張ってください！🚀