YouTube Clone - Complete Project Documentation

Table of Contents

---

1. Project Overview

Introduction

This is a fully functional YouTube clone built with modern React technologies. The project replicates the core features and UI of YouTube, providing a seamless video browsing and watching experience.

Purpose

- Learning modern React development practices
- Demonstrating state management with Redux Toolkit
- Showcasing responsive design with Tailwind CSS
- Implementing data fetching with TanStack Query

Key Highlights

- **60+ mock videos** with real playable content
- **8 channels** with profiles
- **15 categories** for filtering
- **Dark/Light mode** with persistence
- **Responsive design** for all devices
- **Infinite scroll** for video feeds

---

2. Features

Core Features

Homepage

- Video feed grid with thumbnails
- Category-based filtering chips

- Infinite scroll for loading more videos
- Responsive grid layout (1-4 columns)

Video Player

- Full-featured video player with native controls
- Play/Pause, Volume, Seek, Fullscreen
- Progress tracking (saved to history)
- Resume from where you left off

Search

- Search bar with autocomplete suggestions
- Debounced input (300ms delay)
- Recent search history
- Search results page

Navigation

- Collapsible sidebar
- Mobile bottom navigation
- Breadcrumb navigation

User Features

Watch History

- Automatically tracks watched videos
- Shows progress percentage
- Clear all history option

Watch Later

- Save videos to watch later
- Manage playlist
- Clear all option

Liked Videos

- Like videos while watching
- Dedicated liked videos page
- Unlike functionality

Subscriptions

- Subscribe to channels
- Subscription count tracking
- Unsubscribe option

Theme

- Dark mode / Light mode toggle
- Persisted to localStorage
- System preference detection

---

3. Tech Stack

Frontend Framework

Technology	Version	Purpose
React	19.0.0	UI Framework
React DOM	19.0.0	DOM Rendering

Routing

Technology	Version	Purpose
React Router DOM	7.1.1	Client-side routing

State Management

Technology	Version	Purpose
Redux Toolkit	2.3.0	Global state
React Redux	9.2.0	React bindings
TanStack Query	5.62.0	Server state

Styling

Technology	Version	Purpose
Tailwind CSS	3.4.17	Utility CSS
PostCSS	8.4.49	CSS processing
Autoprefixer	10.4.20	Vendor prefixes

HTTP & Media

Technology	Version	Purpose
Axios	1.7.9	HTTP client
react-player	2.16.0	Video playback
react-icons	5.4.0	Icon library

---

4. Project Structure

```
youtube-clone-react/
│
├── public/
```

```
│   ├── index.html          # HTML template
│   ├── manifest.json       # PWA manifest
│   └── robots.txt          # Robots file
│
├── src/
│   ├── api/                # API layer
│   │   ├── axios.js        # Axios instance
│   │   ├── videoApi.js     # API functions
│   │   └── index.js        # Exports
│   │
│   ├── components/         # UI Components
│   │   ├── CategoryBar/
│   │   ├── ChannelCard/
│   │   ├── CommentSection/
│   │   ├── MobileBottomNav/
│   │   ├── Navbar/
│   │   ├── SearchBar/
│   │   ├── Sidebar/
│   │   ├── SkeletonLoader/
│   │   ├── SuggestedVideos/
│   │   ├── VideoCard/
│   │   ├── VideoGrid/
│   │   ├── VideoPlayer/
│   │   └── index.js
│   │
│   ├── data/
│   │   └── mockData.js     # Mock data
│   │
│   ├── hooks/              # Custom hooks
│   │   ├── useCategories.js
│   │   ├── useChannel.js
│   │   ├── useClickOutside.js
│   │   ├── useComments.js
│   │   ├── useIntersectionObserver.js
│   │   ├── useSearch.js
│   │   ├── useVideos.js
│   │   └── index.js
│   │
│   ├── layouts/
│   │   ├── MainLayout.jsx
│   │   ├── WatchLayout.jsx
│   │   └── index.js
│   │
│   ├── pages/
│   │   ├── Channel/
│   │   ├── History/
│   │   ├── Home/
│   │   ├── LikedVideos/
│   │   ├── NotFound/
│   │   ├── Search/
│   │   ├── Trending/
│   │   ├── Watch/
│   │   ├── WatchLater/
│   │   └── index.js
│   │
│   ├── store/
│   │   ├── slices/
│   │   │   ├── historySlice.js
│   │   │   ├── likedVideosSlice.js
│   │   │   ├── searchSlice.js
│   │   │   ├── sidebarSlice.js
│   │   │   ├── subscriptionsSlice.js
│   │   │   ├── themeSlice.js
│   │   │   └── watchLaterSlice.js
│   │   └── index.js
│   │
│   ├── utils/
│   │   └── helpers.js
│   │
│   ├── App.js
│   ├── index.css
│   └── index.js
│
├── docs/                   # Documentation
│   ├── ARCHITECTURE.md
│   ├── API.md
│   ├── COMPONENTS.md
│   ├── SETUP.md
│   └── FULL_DOCUMENTATION.md
│
├── .eslintrc.js
├── .gitignore
├── LICENSE
├── package.json
├── postcss.config.js
├── tailwind.config.js
└── README.md
```

---

5. Installation Guide

Prerequisites

- Node.js v18.0.0 or higher
- npm v9.0.0 or higher
- Git

Step-by-Step Installation

1. Clone Repository

```
git clone https://github.com/yourusername/youtube-clone-react.git
```

```
cd youtube-clone-react
```

2. Install Dependencies

```
npm install
```

3. Start Development Server

```
npm start
```

4. Open in Browser

```
http://localhost:3000
```

Production Build

```
npm run build
```

Output will be in the `build` folder.

Deployment Options

- **Vercel**: `vercel deploy`
- **Netlify**: `netlify deploy --prod`
- **GitHub Pages**: `npm run deploy`

---

6. Architecture

Overview Diagram

```
┌──────────────────────────────────────────────────────────────┐
│                         Application                           │
├──────────────────────────────────────────────────────────────┤
│                                                               │
│  ┌────────────────────────────────────────────────────────┐  │
│  │                    Presentation Layer                    │ │
│  │                                                          │ │
│  │   Pages    →    Layouts    →    Components    →    Hooks │ │
│  │                                                          │ │
│  └──────────────────────────┬─────────────────────────────┘  │
│                             │                                 │
│  ┌──────────────────────────┴─────────────────────────────┐  │
│  │                       State Layer                        │ │
│  │                                                          │ │
│  │        Redux Store              TanStack Query           │ │
│  │       (Client State)           (Server State)           │ │
│  │                                                         │ │
│  └─────────────────────────┬───────────────────────────────┘ │
│                            │                                 │
│  ┌─────────────────────────┴───────────────────────────────┐ │
│  │                       Data Layer                        │ │
│  │                                                         │ │
│  │        API Layer (Axios)         Mock Data              │ │
│  │                                                         │ │
│  └─────────────────────────────────────────────────────────┘ │
│                                                              │
└──────────────────────────────────────────────────────────────┘
```

Data Flow

Reading Data

```
User Action → Component → Hook → TanStack Query → API → Data
                                        ↓
                                   Cache
                                        ↓
                               Component Update
```

Writing Data

```
User Action → Component → Dispatch → Redux Slice → localStorage
                                           ↓
                                    State Update
                                           ↓
                                  Component Update
```

---

7. Components Reference

Layout Components

Component	File	Description
MainLayout	`layouts/MainLayout.jsx`	Navbar + Sidebar + Content
WatchLayout	`layouts/WatchLayout.jsx`	Navbar + Content

Navigation Components

Component	File	Description
Navbar	`components/Navbar/`	Top navigation
Sidebar	`components/Sidebar/`	Side navigation

SearchBar	`components/SearchBar/`	Search with autocomplete
MobileBottomNav	`components/MobileBottomNav/`	Mobile navigation
CategoryBar	`components/CategoryBar/`	Category filters

Video Components

Component	File	Description
VideoCard	`components/VideoCard/`	Video thumbnail card
VideoGrid	`components/VideoGrid/`	Grid of video cards
VideoPlayer	`components/VideoPlayer/`	Video player
SuggestedVideos	`components/SuggestedVideos/`	Related videos

Interactive Components

Component	File	Description
CommentSection	`components/CommentSection/`	Comments display
ChannelCard	`components/ChannelCard/`	Channel info card

Utility Components

Component	File	Description
SkeletonLoader	`components/SkeletonLoader/`	Loading placeholders

---

8. State Management

Redux Slices

Slice	Purpose	Persistence
themeSlice	Dark/light mode	localStorage
sidebarSlice	Sidebar state	No
searchSlice	Search query	localStorage
historySlice	Watch history	localStorage
watchLaterSlice	Watch later	localStorage
likedVideosSlice	Liked videos	localStorage
subscriptionsSlice	Subscriptions	localStorage

TanStack Query Keys

```
['videos', { category, page }]   // Video list
['video', videoId]               // Single video
['channel', channelId]           // Channel info
['comments', videoId]            // Comments
['search', query]                // Search results
['suggestions', query]           // Autocomplete
['categories']                   // Categories
```

Configuration

```
// Query Client Config
{
  defaultOptions: {
    queries: {
      refetchOnWindowFocus: true,
      retry: 1,
      staleTime: 5 * 60 * 1000  // 5 minutes
    }
  }
}
```

---

9. API Reference

Video APIs

getVideos

```
getVideos({ pageParam, category, limit })
// Returns: { videos, nextPage, totalCount }
```

getVideoById

```
getVideoById(videoId)
// Returns: Video object
```

getRelatedVideos

```
getRelatedVideos(videoId, limit)
// Returns: Video array
```

getTrendingVideos

```
getTrendingVideos(limit)
// Returns: Video array (sorted by views)
```

Search APIs

searchVideos

```
searchVideos(query, limit)
// Returns: Video array
```

getSearchSuggestions

```
getSearchSuggestions(query)
// Returns: String array
```

Channel APIs

getChannelById

```
getChannelById(channelId)
// Returns: Channel object
```

getChannelVideos

```
getChannelVideos(channelId, limit)
// Returns: Video array
```

Comments APIs

getComments

```
getComments(videoId)
// Returns: Comment array
```

addComment

```
addComment(videoId, comment)
// Returns: New comment object
```

---

10. Styling Guide

Tailwind Configuration

Custom Colors

```
colors: {
  youtube: {
    red: '#FF0000',
    dark: '#0F0F0F',
    gray: {
      50: '#F9F9F9',
      100: '#F1F1F1',
      200: '#E5E5E5',
      300: '#D3D3D3',
      400: '#AAAAAA',
      500: '#717171',
      600: '#606060',
      700: '#3D3D3D',
      800: '#282828',
      900: '#0F0F0F'
    }
  }
}
```

Responsive Breakpoints

```
screens: {
  'sm': '640px',
  'md': '768px',
  'lg': '1024px',
  'xl': '1280px',
  '2xl': '1536px'
}
```

Dark Mode

Dark mode is class-based:

```
<html class="dark">
```

Usage in components:

```
<div className="bg-white dark:bg-youtube-dark">
```

---

11. Future Enhancements

Planned Features

- □ Real YouTube API integration
- □ User authentication (OAuth)
- □ Video upload functionality
- □ Playlist creation and management
- □ YouTube Shorts support
- □ Live streaming
- □ Push notifications
- □ Picture-in-Picture mode
- □ Keyboard shortcuts
- □ Video quality selection
- □ Subtitles/Captions support
- □ Share functionality
- □ Mini player
- □ Comments with replies
- □ Channel analytics

Performance Improvements

- □ Code splitting with lazy loading
- □ Image optimization
- □ Service worker for offline support
- □ Bundle size optimization

Testing

- □ Unit tests with Jest
- □ Integration tests with Testing Library
- □ E2E tests with Cypress

Contact & Support

Author

Your Name

Repository

https://github.com/yourusername/youtube-clone-react

License

MIT License

Documentation generated for YouTube Clone React v0.1.0 Last updated: December 2024