### **1. Introduction**

### Cryptowatcher is a web application that provides information about cryptocurrencies, including real-time data on prices, market capitalization, and other relevant metrics. This documentation outlines the project structure, key components, and pages to help developers understand and contribute to the project.

### **2. Getting Started**

### InstallationTo get started with the Cryptowatcher project, follow these steps:

* Clone the repository
* Install dependencies: npm install
* Start the development server: npm run dev

#### Project StructureThe project is structured as follows:

* components: Contains React components used in the application.
* assets: Houses SVG icons and other static assets.
* pages: Defines the different pages of the application.
* styles: Includes styling constants used throughout the application.

### **3. Components**

#### Header

#### The Header component represents the top navigation bar of the application, displaying the Cryptocapwatcher logo and a badge.

// Example usage

import Header from "@/components/Header";

const MyComponent = () => {

return (

<div>

<Header />

{/\* Additional content \*/}

</div>

);

};

#### CryptoTable

#### The CryptoTable component fetches cryptocurrency data from an API and displays it in a table format.

// Example usage

import CryptoTable from "@/components/cryptotable/CryptoTable";

const MyComponent = () => {

return (

<div>

<CryptoTable />

{/\* Additional content \*/}

</div>

);

};

#### CryptoTableHeader

#### The CryptoTableHeader component represents the header row of the cryptocurrency table.

// Example usage

import CryptoTableHeader from "@/components/cryptotable/CryptoTableHeader";

const MyComponent = () => {

return (

<table>

<CryptoTableHeader />

{/\* Additional table rows \*/}

</table>

);

};

#### CryptoTableRow

#### The CryptoTableRow component represents a single row in the cryptocurrency table.

// Example usage

import CryptoTableRow from "@/components/cryptotable/CryptoTableRow";

const MyComponent = () => {

return (

<table>

<tbody>

<CryptoTableRow num={1} name="Bitcoin" price="$50,000" market\_cap="$1 trillion" circulating\_supply="18.7 million" />

{/\* Additional table rows \*/}

</tbody>

</table>

);

};

### **4. Pages**

### Home

### The Home page is the main landing page of the application, displaying the header and cryptocurrency table.

#### Info

The Info page displays detailed information about a specific cryptocurrency based on the selected coin.

### **5. Styles**

### The styles object contains various styling constants used throughout the application. These constants can be reused for consistent styling.

### **6. Manifest Files**

The manifest.json file defines metadata about the Cryptowatcher extension. It includes information such as the extension name, version, description, and the default popup HTML file.

This documentation provides an overview of the Cryptowatcher project, its components, and how to get started. Developers are encouraged to explore the codebase and contribute to the project's growth.