

Building a Crypto Market App with Next.js

Introduction

In this ebook, we will walk through the creation of a cryptocurrency market application using modern web development technologies. The app displays the top 20 cryptocurrencies (excluding stablecoins) by market capitalization and shows relevant market data using a clean, dark-themed interface.

Technologies Used

1. Next.js (App Router)

Next.js is a popular React framework that offers server-side rendering, static site generation, and API routes out-of-the-box. We use the App Router (new routing system) introduced in Next.js 13+.

2. TypeScript

TypeScript is a statically typed superset of JavaScript that helps catch errors early through type checking, making our codebase more maintainable and scalable.

3. SWR

SWR is a data fetching library from Vercel that makes it easy to handle caching, revalidation, focus tracking, and more in React.

4. Bootstrap (Dark Theme)

Bootstrap is a popular CSS framework. We use it for styling components and enable the dark theme using Bootstrap's data attributes.

5. CoinGecko API

We fetch real-time market data using CoinGecko's public API. It requires no authentication and is well-suited for public crypto dashboards.

6. Vercel

Vercel is the platform behind Next.js. We use it to deploy the application with one-click GitHub integration.

Features Implemented

- List of top 20 cryptocurrencies by market cap
- Exclusion of stablecoins
- Display of:
- Logo

- Symbol
 - Name
 - Current price
 - Price variation (24h and 7d)
 - Responsive dark-themed UI using Bootstrap
 - Live data fetching with SWR and CoinGecko API
-

Step-by-Step Guide

Step 1: Create the Next.js Project

```
npx create-next-app@latest crypto-market-app --ts
cd crypto-market-app
```

Step 2: Install Dependencies

```
npm install bootstrap swr
```

Step 3: Configure Global Layout

Edit `app/layout.tsx`:

```
import 'bootstrap/dist/css/bootstrap.min.css';
import './globals.css';

export const metadata = {
  title: 'Top 20 Cryptos',
  description: 'Top 20 Market Cap Cryptocurrencies (Excluding Stablecoins)',
};

export default function RootLayout({ children }: { children:
  React.ReactNode }) {
  return (
    <html lang="en" data-bs-theme="dark">
      <body className="bg-dark text-light">{children}</body>
    </html>
  );
}
```

Step 4: Create Utility Functions

`lib/fetcher.ts`

```
export const fetcher = (url: string) => fetch(url).then(res => res.json());
```

lib/filterStablecoins.ts

```
const stablecoinSymbols = ['USDT', 'USDC', 'DAI', 'TUSD', 'BUSD', 'FDUSD', 'GUSD'];

export const excludeStablecoins = (coins: any[]) =>
  coins.filter((coin) => !
    stablecoinSymbols.includes(coin.symbol.toUpperCase()));
```

Step 5: Create the Home Page

app/page.tsx

```
'use client';

import useSWR from 'swr';
import { fetcher } from '@lib/fetcher';
import { excludeStablecoins } from '@lib/filterStablecoins';
import Image from 'next/image';

export default function Home() {
  const { data, error, isLoading } = useSWR(
    'https://api.coingecko.com/api/v3/coins/markets?vs_currency=usd&order=market_cap_desc&per_page=30&page=1&sparkline=false&price_change_percenta',
    fetcher,
    { refreshInterval: 60000 }
  );

  if (error) return <p className="text-danger">Failed to load data</p>;
  if (isLoading) return <p>Loading...</p>;

  const coins = excludeStablecoins(data ?? []).slice(0, 20);

  return (
    <div className="container py-5">
      <h1 className="mb-4">Top 20 Cryptocurrencies (No Stablecoins)</h1>
      <table className="table table-dark table-striped table-hover">
        <thead>
          <tr>
            <th>Logo</th>
            <th>Name</th>
            <th>Symbol</th>
            <th>Price (USD)</th>
            <th>Change (24h)</th>
            <th>Change (7d)</th>
          </tr>
        </thead>
        <tbody>
          {coins.map((coin) => (
```

```

        <tr key={coin.id}>
          <td>
            <Image src={coin.image} alt={coin.name} width={24}
height={24} />
          </td>
          <td>{coin.name}</td>
          <td>{coin.symbol.toUpperCase()}</td>
          <td>${coin.current_price.toLocaleString()}</td>
          <td className={coin.price_change_percentage_24h > 0 ? 'text-
success' : 'text-danger'}>
            {coin.price_change_percentage_24h.toFixed(2)}%
          </td>
          <td className={coin.price_change_percentage_7d_in_currency >
0 ? 'text-success' : 'text-danger'}>
            {coin.price_change_percentage_7d_in_currency.toFixed(2)}%
          </td>
        </tr>
      </tbody>
    </table>
  </div>
);
}

```

Step 6: Allow External Images

Edit `next.config.js`:

```

/** @type {import('next').NextConfig} */
const nextConfig = {
  images: {
    domains: ['assets.coingecko.com'],
  },
};

module.exports = nextConfig;

```

Step 7: Run the App Locally

```
npm run dev
```

Visit <http://localhost:3000>

Deploy to Vercel

1. Push your code to a GitHub repository.
2. Go to <https://vercel.com> and log in.

3. Click **New Project** and import your GitHub repo.
 4. Vercel will auto-detect the Next.js framework.
 5. Click **Deploy**.
 6. Done! You'll get a live URL like: `https://crypto-market-app.vercel.app`
-

Conclusion

This app showcases how to build a live cryptocurrency dashboard using modern frontend tools. With Next.js and SWR, you can handle data-fetching and UI elegantly. Vercel makes deployment a breeze. With this foundation, you can extend the app to support charts, search, filters, and even user authentication.

Happy coding!