Cryptoverse

Cryptocurrency dashboard

TEAM ID : SWTID1741264814149484

TEAM LEADER : Raghul kannan S
EMAIL ID : raghulkannan005@gmail.com
ROLE: Project Manager

TEAM MEMBER: MANIKANDAN M
EMAIL ID : ss1569193@gmail.com
ROLE: Assistant Project Manager

TEAM MEMBER : ABIRAJ D
EMAIL ID : abirajrio@gmail.com
ROLE: Documentation

TEAM MEMBER: Jeevan J
EMAIL ID : sambzia2004@gmail.com
ROLE: Demo video maker

TEAM MEMBER: Ramachandran S
EMAIL ID : duke3456harish@gmail.com
ROLE: Project Tester

==

CRYPTOVERSE
A COMPREHENSIVE CRYPTOCURRENCY PLATFORM
PROJECT DOCUMENTATION

==

TABLE OF CONTENTS

----

1. INTRODUCTION

Cryptoverse is a comprehensive web application designed to provide users with real-time cryptocurrency prices, market cap rankings, and the latest crypto news. The application is built with a robust backend server using Node.js and Express, and a dynamic frontend client using React and Vite.

This documentation provides a detailed overview of the project's architecture, features, implementation details, and setup instructions.

2. FEATURES

2.1 User Authentication

- Registration system for new user account creation
- Secure login functionality for existing users
- Logout mechanism
- JWT token verification for secure access control

2.2 Cryptocurrency Data

- Real-time cryptocurrency price tracking
- Market capitalization metrics
- Trading volume information
- Performance-optimized data retrieval through caching

2.3 Cryptocurrency Details

- Historical price data visualization
- Comprehensive market statistics
- Detailed coin information
- Historical performance tracking

2.4 Global Cryptocurrency Stats

- Total cryptocurrency count
- Exchange platform metrics
- Global market capitalization
- 24-hour trading volume aggregation
- Total markets tracking

2.5 Latest Cryptocurrency News

- Category-based news organization
- Sentiment analysis on news articles
- Cached news data for optimized performance

2.6 Contact Form

- User inquiry submission system
- Email and phone contact information

==

3. TECHNICAL ARCHITECTURE

3.1 Backend Technologies

- Node.js: JavaScript runtime environment
- Express: Web application framework
- MongoDB (Mongoose): NoSQL database system
- JWT (jsonwebtoken): Token-based authentication
- bcryptjs: Password encryption library
- axios: HTTP client for API requests
- dotenv: Environment variable management
- helmet: HTTP security middleware
- express-rate-limit: Request limitation middleware
- node-cache: In-memory caching system

3.2 Backend Project Structure

Controllers:
- authController.js: Authentication process management
- cryptoController.js: Cryptocurrency data handling

Middlewares:
- authMiddleware.js: Route protection via JWT verification
- cacheMiddleware.js: API response caching
- errorMiddleware.js: Error handling and response formatting

Models:
- userModel.js: MongoDB user schema definition

Routes:
- authRoutes.js: Authentication endpoint definitions
- cryptoRoutes.js: Cryptocurrency endpoint definitions

Utils:
- db.js: MongoDB connection management

Configuration:
- .env.example: Environment variable template
- vercel.json: Vercel deployment configuration

3.3 Frontend Technologies

- React: UI component library
- Vite: Frontend build tool
- Tailwind CSS: Utility-first styling framework
- Axios: HTTP client
- React Router: Navigation system
- Chart.js: Data visualization library
- Moment.js: Date manipulation utility
- Millify: Number formatting library
- React Icons: Icon component library

3.4 Frontend Project Structure

Components:
- Homepage.jsx: Main dashboard display
- Cryptocurrencies.jsx: Cryptocurrency listing with search
- CryptoDetails.jsx: Individual cryptocurrency information
- News.jsx: Cryptocurrency news display
- Contact.jsx: User inquiry form
- About.jsx: Platform information
- Login.jsx: Authentication interface
- Register.jsx: User registration interface
- Navbar.jsx: Navigation component
- Spinner.jsx: Loading indicator

Context:
- AuthContext.jsx: Authentication state management

Hooks:
- useAuth.js: Authentication state custom hook

Services:
- axiosConfig.js: HTTP client configuration
- cryptoApi.js: Cryptocurrency data service
- newsApi.js: News data service

Styles:
- index.css: Tailwind CSS implementation

Configuration:
- vite.config.js: Build tool settings
- tailwind.config.js: CSS framework configuration

Entry Point:
- main.jsx: Application initialization

===

4. ENVIRONMENT CONFIGURATION

The application requires the following environment variables:

Backend Configuration:

- MONGO_URI: MongoDB connection string
- PORT: Server port number
- COINLAYER_API_KEY: CoinLayer API authentication
- NEWSDATA_API_KEY: NewsData API authentication
- JWT_SECRET: Token encryption key

Frontend Configuration:

- VITE_BACKEND_URL: Backend API endpoint URL

===

5. DEPLOYMENT PROCEDURE

The application is configured for deployment on Vercel using the
vercel.json
configuration file. This provides seamless continuous deployment from the
repository.

===

6. INSTALLATION AND SETUP GUIDE

----

1. Repository Acquisition
   ```
   $ git clone https://github.com/Raghulkannan005/cryptoverse.git
   $ cd cryptoverse
   ```

2. Dependency Installation
   ```
   $ cd backend
   $ npm install
   $ cd ../frontend
   $ npm install
   ```

3. Environment Configuration
   - Copy .env.example to .env in both backend and frontend directories
   - Configure the environment variables with appropriate values

4. Backend Server Initialization
   ```
   $ cd backend
   $ npm run dev
   ```

5. Frontend Development Server Initialization
   ```
   $ cd frontend
   $ npm run dev
   ```

6. Access the application at http://localhost:5173

===

CONCLUSION

Cryptoverse provides a comprehensive solution for cryptocurrency enthusiasts to track market trends, view detailed coin information, and stay updated with the latest news in the crypto space. The application demonstrates effective implementation of modern web development practices including responsive design, secure authentication, and optimized data retrieval.

==