Open Source GitHub Project Explorer

Introduction

The Open Source GitHub Project Explorer is a dynamic web-based dashboard designed to explore and analyze trending repositories on GitHub. It provides developers, students, and open-source enthusiasts with an intuitive interface to discover repositories based on programming languages, popularity, and update activity. This project emphasizes user interactivity, analytics, and efficient data visualization.

Abstract

This project leverages the GitHub REST API to fetch real-time repository data and present it in a user-friendly interface. It allows filtering repositories by criteria such as stars, language, and last updated date. The dashboard integrates charts using Chart.js to visualize repository statistics, including contributions and issues. Additional features like bookmarking and note-taking provide a personalized experience for users exploring repositories.

Tools Used

React.js – For building an interactive, component-based frontend.
GitHub REST API – To fetch and manage repository data dynamically.
Chart.js – For rendering graphical analytics on repository performance.
Tailwind CSS – To design a modern and responsive UI efficiently.

Implementation Steps

1. Setup a React app using Vite or Create React App.
2. Integrated GitHub REST API to fetch trending repositories and metadata.
3. Implemented sorting and filtering options based on stars, language, and update date.
4. Developed interactive charts using Chart.js for repository metrics.
5. Designed a responsive interface using Tailwind CSS.
6. Added features for bookmarking and taking custom notes.
7. Tested and deployed the dashboard online.

Conclusion

The Open Source GitHub Project Explorer demonstrates how APIs and modern frontend frameworks can be combined to create analytical and engaging tools that promote open-source discovery and collaboration.

Custom CMS for Blog Publishing

Introduction

The Custom CMS for Blog Publishing project focuses on building a lightweight, headless content management system (CMS) to efficiently handle blog content. It empowers administrators to create, edit, and manage posts while offering a seamless frontend experience for readers. The system separates content management from presentation, ensuring scalability and flexibility across multiple platforms.

Abstract

This CMS provides a secure and user-friendly way to manage blogs, drafts, and categories using an API-driven backend. It utilizes Express.js and MongoDB for data handling, JWT for authentication, and a React-based frontend integrated with a rich text editor (Quill.js) to enable authors to format and preview their posts dynamically.

Tools Used

Express.js – For building backend APIs and managing server logic.
MongoDB – For storing blog content, drafts, and user data.
JWT (JSON Web Token) – For secure authentication and admin access control.
React.js – To build an interactive, responsive blog frontend.
Quill.js – To implement a feature-rich text editor for blog creation.

Implementation Steps

1. Created backend routes in Express.js for creating, updating, and retrieving blog posts.
2. Integrated MongoDB for efficient content storage and category management.
3. Implemented JWT-based admin authentication for secure CMS access.
4. Developed a React frontend with a Quill.js editor for blog writing.
5. Added functionality for managing drafts, publishing, and editing posts.
6. Built frontend routes to dynamically display blogs by category.
7. Optional: Added a static export feature for SEO optimization.

Conclusion

The Custom CMS for Blog Publishing project demonstrates the creation of a modular and scalable content platform using the MERN stack. By decoupling the backend (content API) from the frontend (React blog site), it achieves flexibility, security, and ease of management for modern digital publishing.