

Example

Description

Created a dynamic, customizable dashboard supporting various widgets to display the issue count for each template, the trend of issue count throughout a certain number of days, etc. using Material-UI components for React. Refactored common properties of widgets such as widget titles, subtitles, and configuration menu wrapper to a shared component, enabling cleaner implementation and easier creation of new widgets.

Built an interface allowing admin users to configure issue templates with customizable input options such as name, type and more. Implemented dynamically rendered issue submission forms based on the conditional rendering of different input options into MUI input components by specified input options.

Created an interface for issue owners of a particular category of issues to view the details of an issue, respond to the issue, and mark keywords that serve as evidence for particular attributes of the issue, enabling future training of machine learning models for automatic attribute classification.

Implemented ExpressJS backend endpoints for retrieving template data and issue-related statistics such as issue counts and attributes. Connected the backend to MongoDB Atlas via Mongoose for better scalability and data management.

Leveraged Faker.js to generate mock data for testing the implementation of the application, ensuring robustness and reliability. Containrized the backend with Docker and deployed to Google Cloud Platform.

Developed a basic authentication mechanism, storing user info in localStorage upon successful login and clears it upon logout. Implemented the frontend routing using React Router and ensured unauthorized users are redirected to an error page.

Keywords

JavaScript, TypeScript, HTML5, CSS3, Python, ReactJS, jQuery, Bootstrap, Material-UI, RESTful API, ExpressJS, NodeJS, Microservices, Figma, Docker, Git, MongoDB, PostgreSQL, MySQL, Amazon Web Service(AWS), Google Cloud Platform(GCP), Vercel