

Feature Specifications

Provides functionality to add all the fields like name,type,description,model repos,feature repos and the users of the projects while adding new-projects to the admin panel.so that it can make it easier to manage, collaborate, and understand the projects, as well as provide a centralized location for users to access them.

Getting Started with Create React App

This project was bootstrapped with [Create React App](#).

Available Scripts

In the project directory, you can run:

Running the App

npm start

Runs the app in the development mode.
Open <http://localhost:3000> to view it in your browser.

The page will reload when you make changes.
You may also see any lint errors in the console.

npm test

Launches the test runner in the interactive watch mode.
See the section about [running tests](#) for more information.

npm run build

Builds the app for production to the `build` folder.

It correctly bundles React in production mode and optimizes the build for the best performance.

The build is minified and the filenames include the hashes.

Your app is ready to be deployed!

See the section about [deployment](#) for more information.

`npm run eject`

Note: this is a one-way operation. Once you `eject`, you can't go back!

If you aren't satisfied with the build tool and configuration choices, you can `eject` at any time. This command will remove the single build dependency from your project.

Instead, it will copy all the configuration files and the transitive dependencies (webpack, Babel, ESLint, etc) right into your project so you have full control over them. All of the commands except `eject` will still work, but they will point to the copied scripts so you can tweak them. At this point you're on your own.

You don't have to ever use `eject`. The curated feature set is suitable for small and middle deployments, and you shouldn't feel obligated to use this feature. However we understand that this tool wouldn't be useful if you couldn't customize it when you are ready for it.

Integration

Request (in props):

- i. Project Name: Choose a clear and concise name for your project.name as a string.if not provided then it will return the error.
- ii. Project Type: Categorize your project according to its type, such as audio streams, live streams, shorts, video, articles, treasury, generatives etc..
- iii. Project Description: Write a brief description of your project that explains its purpose, goals, and potential benefits. Description as a string. If not provided, it should render the default description.
- iv. Model Repository: Create a repository where you can store and manage the models used in your project.

- v. **Feature Repository:** Create a repository where you can store and manage the features used in your project.
- vi. **Data Repository:** Create a repository where you can store and manage the data used in your project.
- vii. **User Access:** Determine who will have access to your project repositories, and set appropriate permissions to ensure that only authorized users can make changes.
- viii. **Collaborators:** The people who will be working on the project, including their roles and responsibilities.
- ix. **API Connectors:** Any external APIs or web services that the project will integrate with, along with documentation on how to use them.

Usage

By including details such as name of the project, project type, description, model repos, and feature repos, it makes it easier to understand what each project is about and what resources are associated with it.

The admin panel can make it easier for users to access and use the projects, as they can find them all in one place.

Adding all the projects to the admin panel with detailed information can make it easier to manage, collaborate, and understand the projects, as well as provide a centralized location for users to access them.