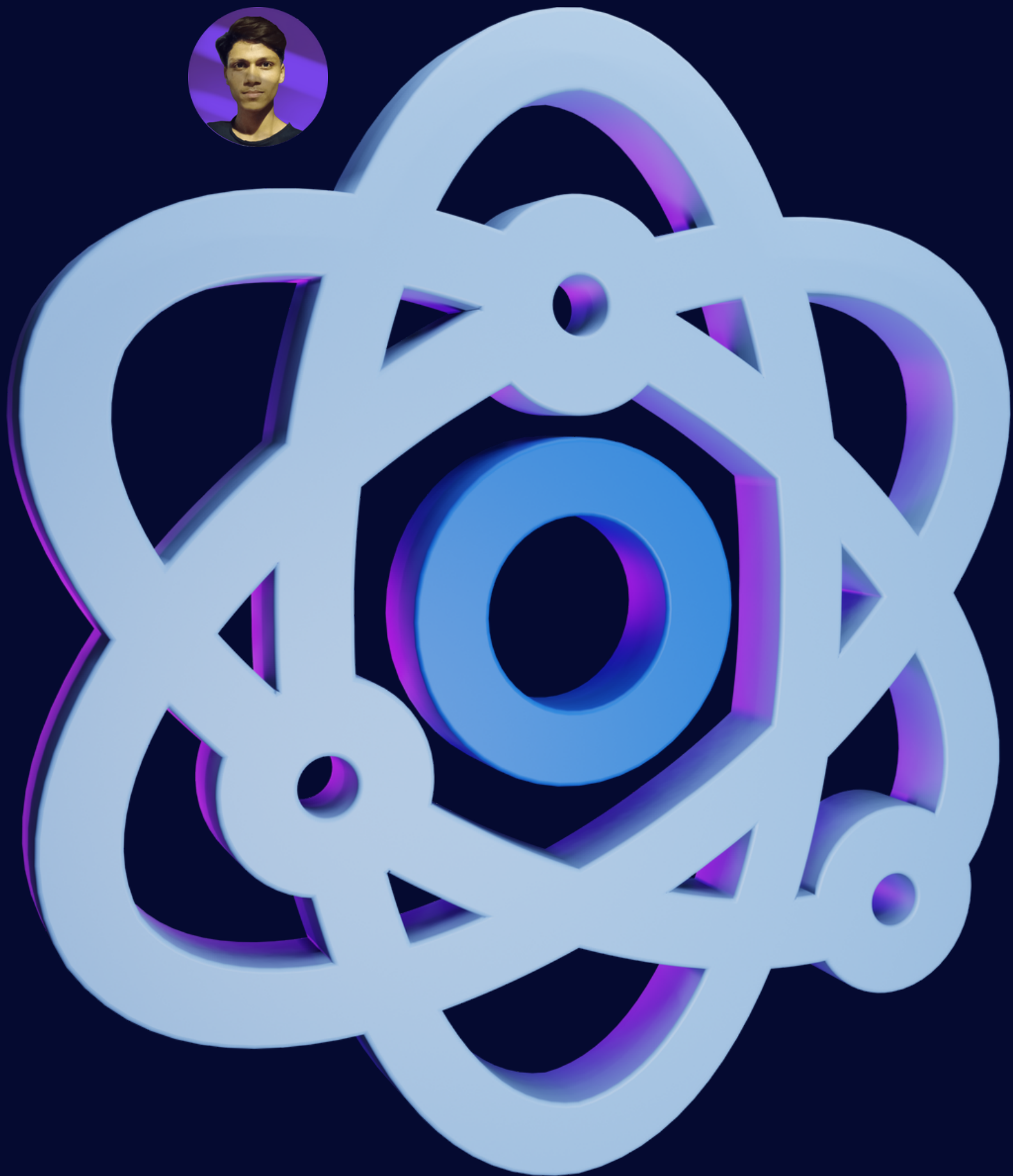


React JS

.env

complete guide



`.env` File | Usage

The `.env` file is commonly used to store environment variables. these variables are configuration values that can be used to customize the behavior of your application depending on the environment it's running in (e.g., development, production, Keys etc).

React applications often use environment variables to store sensitive information or configuration settings.



How to use an `.env` file



Create an `.env` file:

In the root of your React project, create a file named `.env`. You can also have specific environment files like `.env.development`, and `.env.production`, etc., to manage different configurations for different environments.



Add variables

Inside the .env file, you can define variables like this:

```
.env  
  
// With VITE  
VITE_API_KEY=your_api_key  
VITE_API_URL=https://api.example.com  
  
// With Create React App (CRA)  
REACT_APP_API_KEY=your_api_key  
REACT_APP_API_URL=https://api.example.com
```

Note that variable names must start with REACT_APP_ to be recognized by Create React App & VITE _ by Vite.

This prefix is a safety feature to avoid accidentally exposing private keys.



Access the **variables**

In your React components or other parts of your code, you can access these variables like this

```
                                .MyComponent.jsx
// With VITE
const apiKey = import.meta.env.VITE_API_KEY;
const apiUrl = import.meta.env.VITE_API_URL;

// With Create React App (CRA)
const apiKey = process.env.REACT_APP_API_KEY;
const apiUrl = process.env.REACT_APP_API_URL;
```

Remember that these values are available at build time, so if you change them, you'll need to restart your development server or rebuild your application.



Use Variables in configuration

```
const apiUrl = import.meta.env.VITE_API_URL;
const apiKey = import.meta.env.VITE_API_KEY;

const fetchData = async () => {
  const response = await fetch(apiUrl, {
    headers: {
      'Authorization': `Bearer ${apiKey}`,
      'Content-Type': 'application/json'
    }
  });

  if (response.ok) {
    console.log(await response.json());
  } else {
    console.error(`Error! ${response.status}`);
  }
};

export default fetchData; .MyComponent.jsx
```





FOLLOW

