Blogs-App (App displaying posts.)

Overview:

In web apps when you press ‘home’ you are redirected to home page. There is a mechanism known as routing which decides what to display when a user takes any action.

You will learn to implement routing in this blog-app and also to access remote data using axios. Blog-app consists of posts, users and comments (related to blogs). An user can have multiple posts.

There is one user per posts. There can be multiple comments per blog. A user can comment on multiple blogs.

1. Clone Repository

You can clone github repository in two ways:

Visiting repo and cloning via github desktop

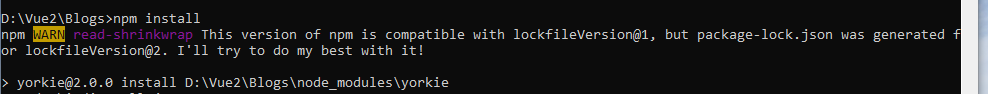
<https://github.com/PreethaKachhadiyaBacancy/Blog-Final-Project>

OR

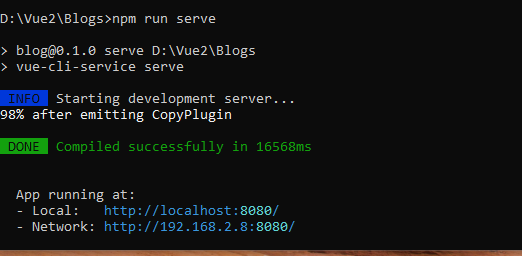
Running this command in the terminal

git clone https://github.com/ PreethaKachhadiyaBacancy/Blog-Final-Project.git

1. Install Dependencies

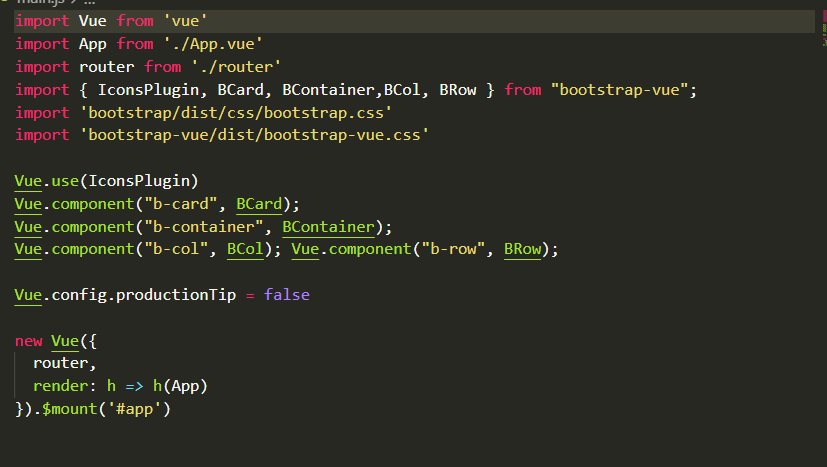


1. Run the app



1. Created UI

Main.js: Here, we import bootstrap-vue and bootstrap-vue.css functionalities to use them in our application to create UI.



Router.js:

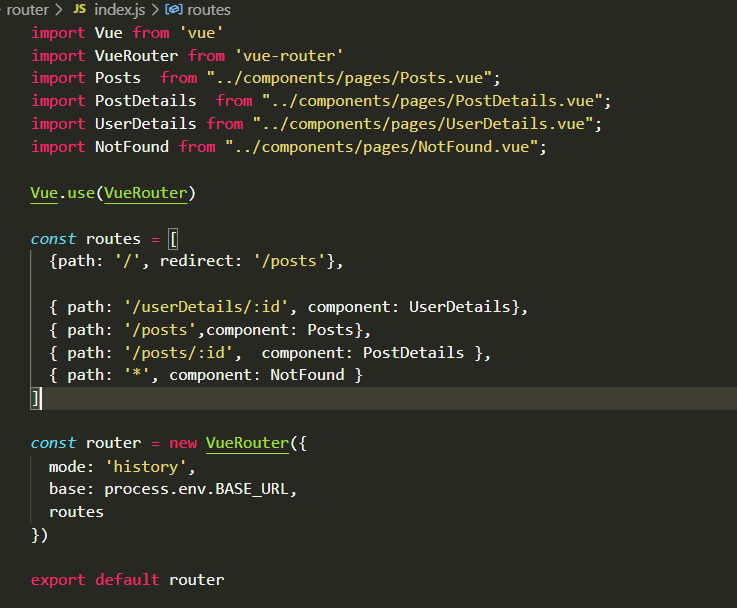
Here, the ‘vue-router’ package is imported to set navigation routes for our components.

There are five routes defined. Each route contains one component. ‘/’ defines home. It will show posts to user.

‘/posts’ shows the posts, thus similar to home route.

‘/posts/:id’ shows post details which includes the post and comments related to it. ‘/userDetails/:id’ shows details of the user.

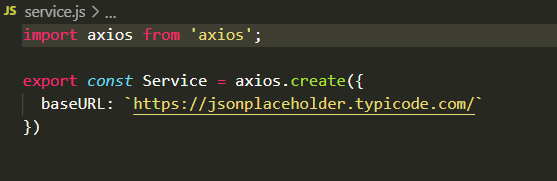
‘\*’ is known as default route. This is a special route which is executed when a route typed by user does not exist in the defined routes.



Service.js:

Data is stored in a remote location, which is accessed by URL. Here, we are importing ‘axios’ functionality to access the remote data.

We are creating an instance of axios using ‘create()’ method where the resource is configured(accessed) via baseURL prop.



UI/BaseComment.vue:

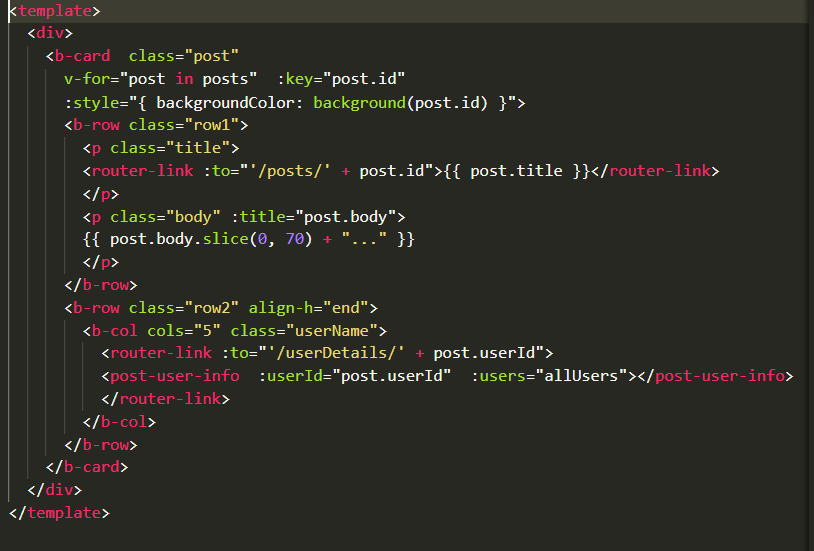
Component shows the comments related to post.



UI/BasePost.vue:

Component displays the post created by user.

‘<router-link>’ tag is used to access routes in VueJS.



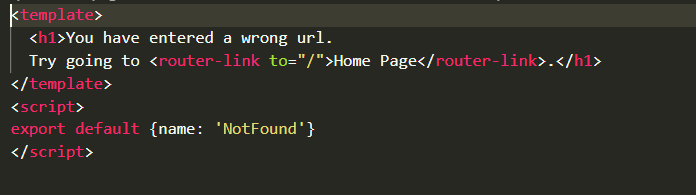
UI/PostUserInfo.vue:

This component displays information of a user related to post.



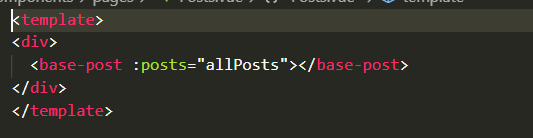
Pages/NotFound.vue:

This page is displayed when user enters an undefined route.



Pages/Posts.vue:

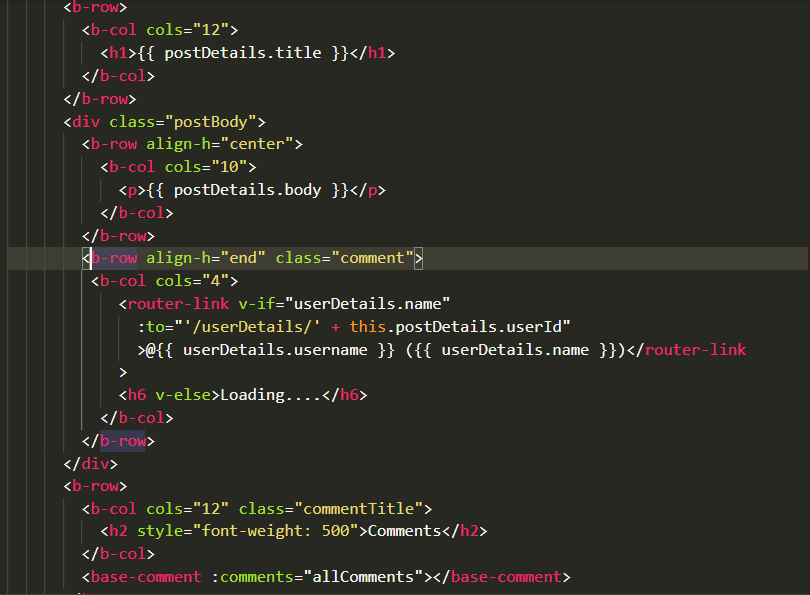
This page shows the details of all posts using ‘base-post’ component.



Pages/PostDetails.vue:

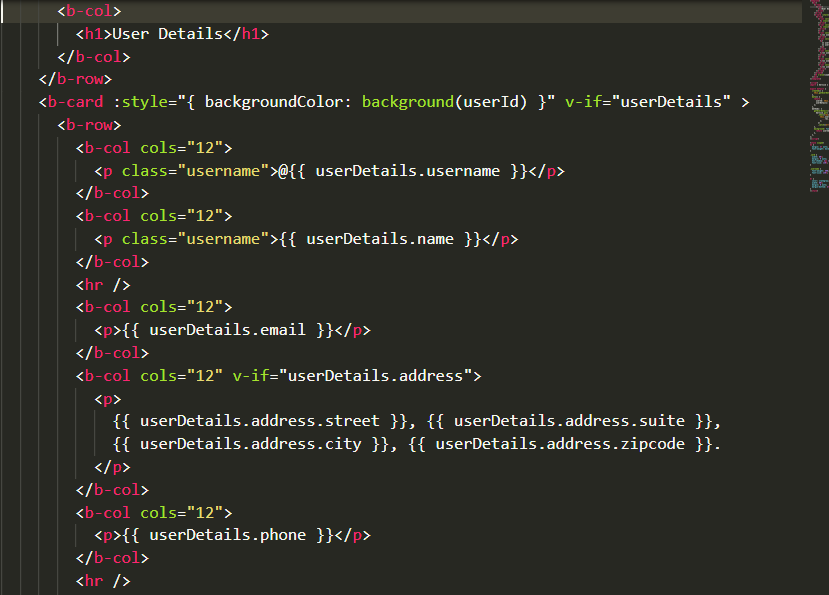
This page shows details of a post and user related to post.

Also displays comments related to posts using ‘base-comment’ component.



Pages/UserDetails.vue:

This page displays details of a user.



1. Blog-App Implementation

Posts.Vue:

In the getAllPosts method, the post data are fetched by ‘get’ method of axios instance from remote URL passed in ‘get’ method.

This data is stored in an array. ‘getAllPosts’ method is called during created hook of the app.



BasePost.vue:

This component contains the UI for displaying the posts. The ‘posts’ prop accepts the data from other components.

This app is loaded first in the browser, as mounted function is called. User data related to posts are loaded because we are fetching user data when app is mounted.

The User details are passed to ‘PostUserInfo’ component.



PostUserInfo.vue:

This component displays user data.

It accepts two props ‘userId’ and ‘users’.

It has two computed properties ‘username’ and ‘name’ which checks details of specific user via ‘userId’ prop.



PostDetails.vue:

getPostDetails method fetches post by its ‘id’, which is stored in postDetails object. This method calls two other methods ‘getComments(fetching comments related to post)’ and ‘getUserDetails(User related to post)’.

BaseComment.vue component displays all the comments. ‘allComments’ array is passed to ‘comments’ props.

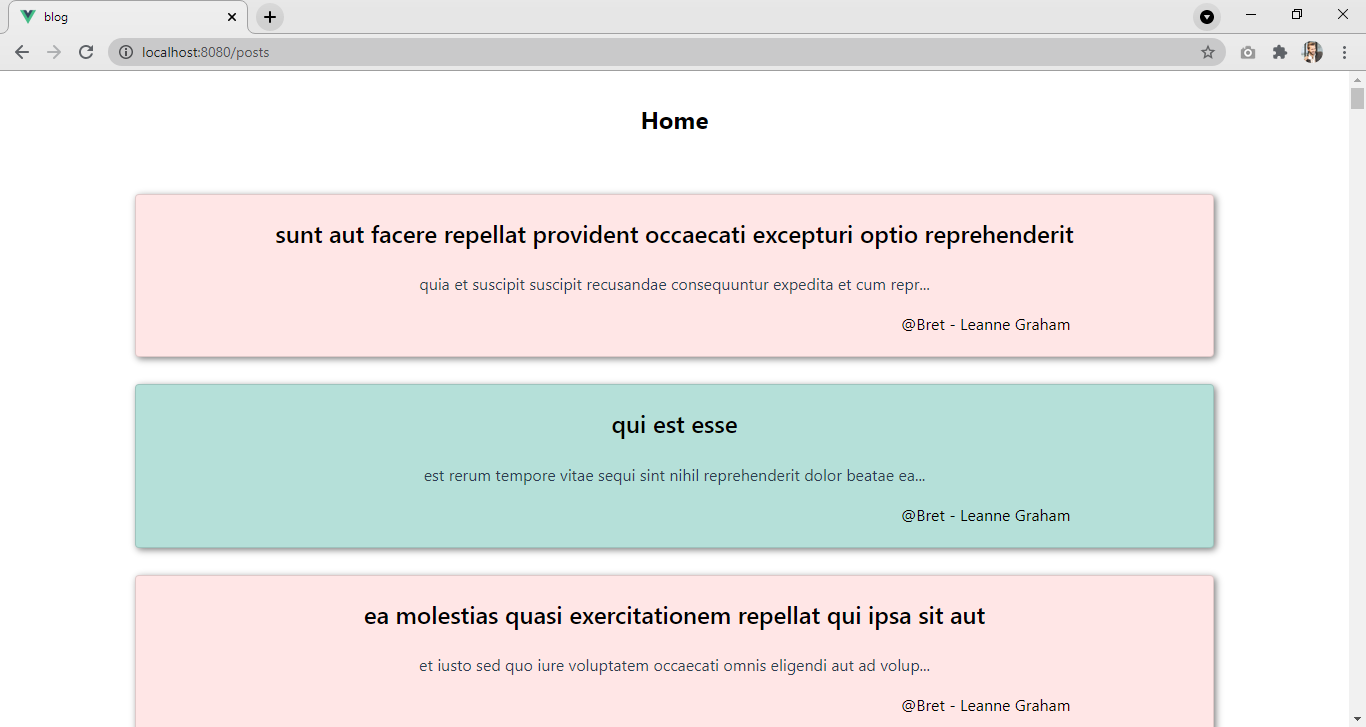


BaseComment.vue:

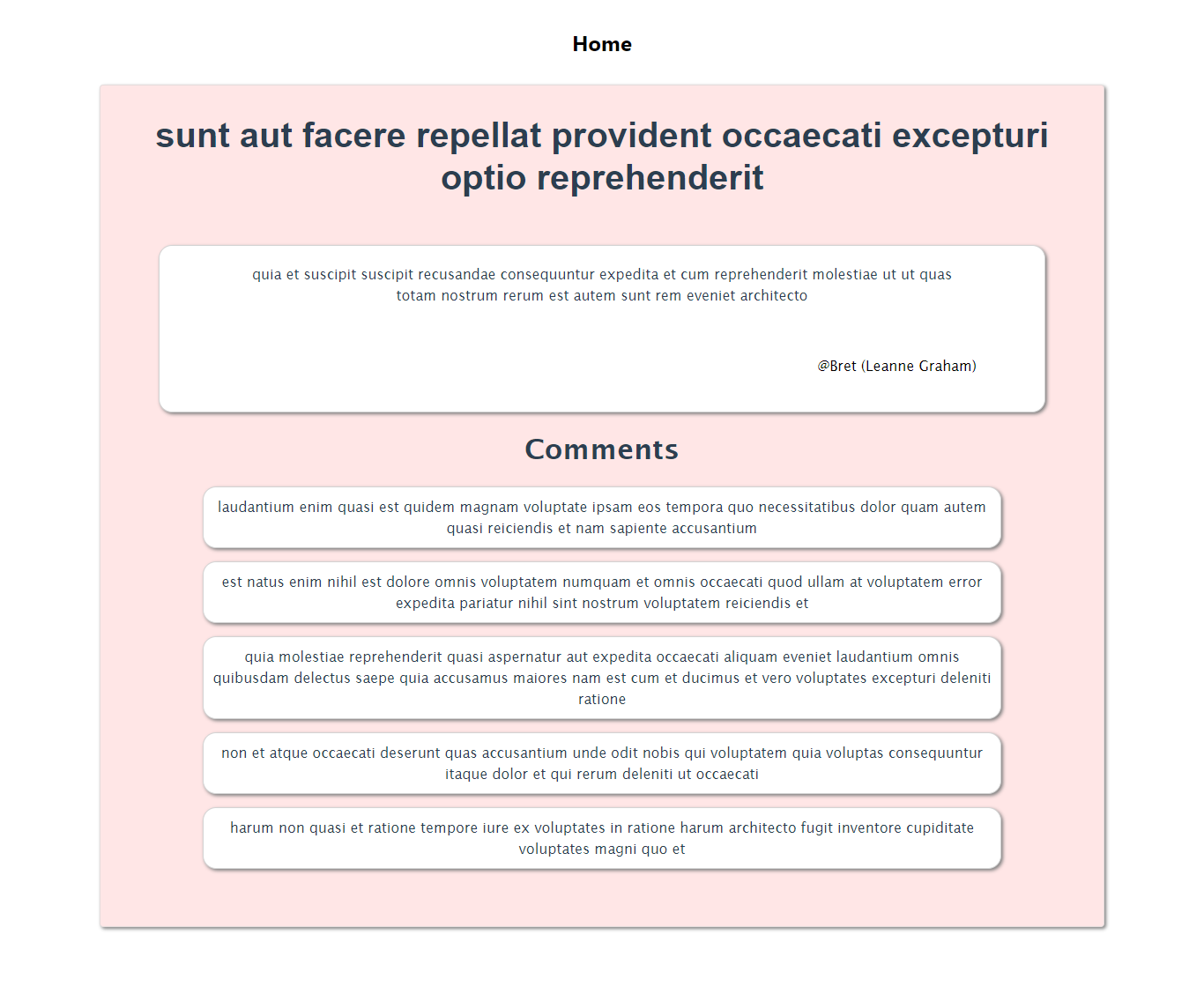


1. Output

Posts:



PostDetails:



UserDetails:

