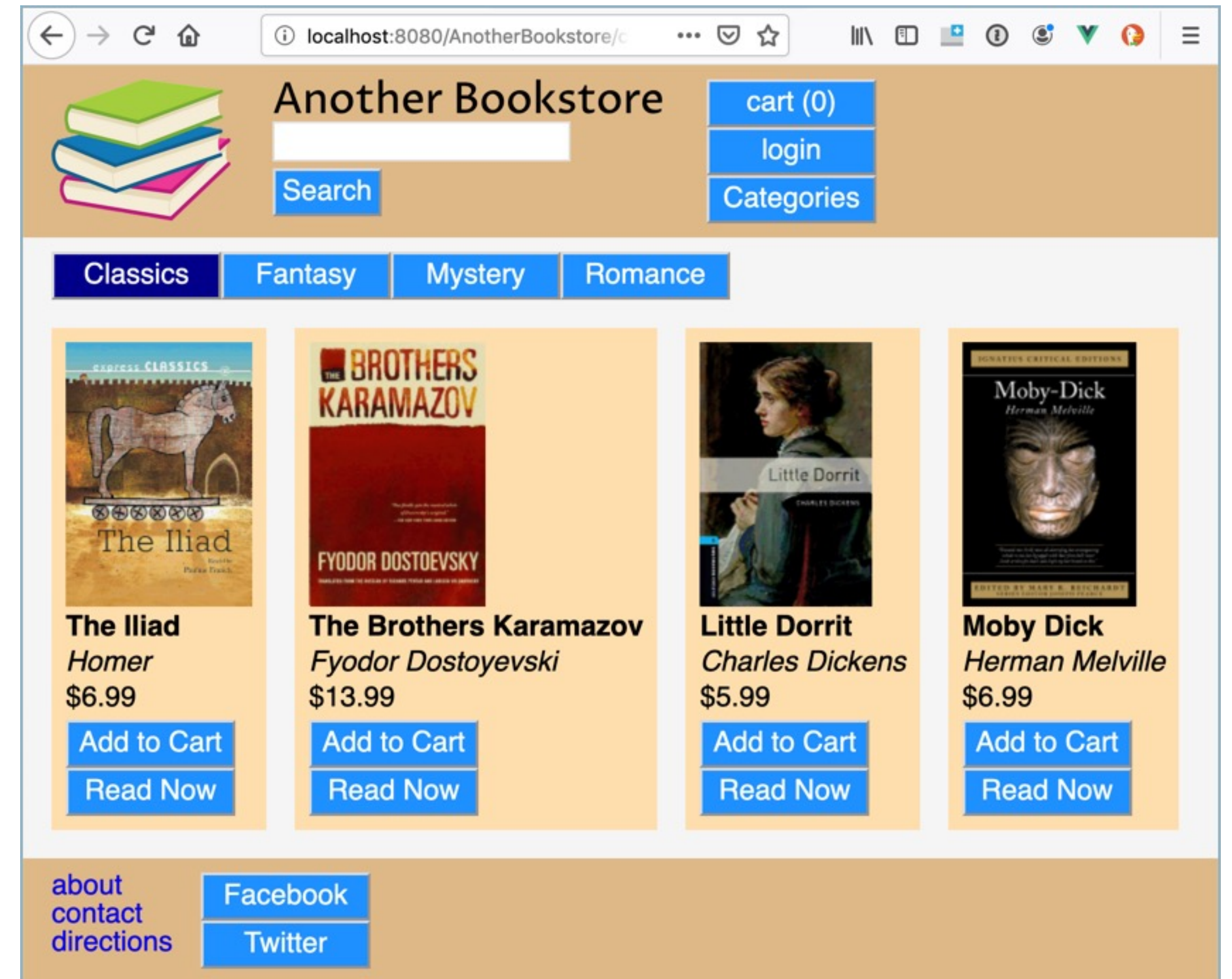
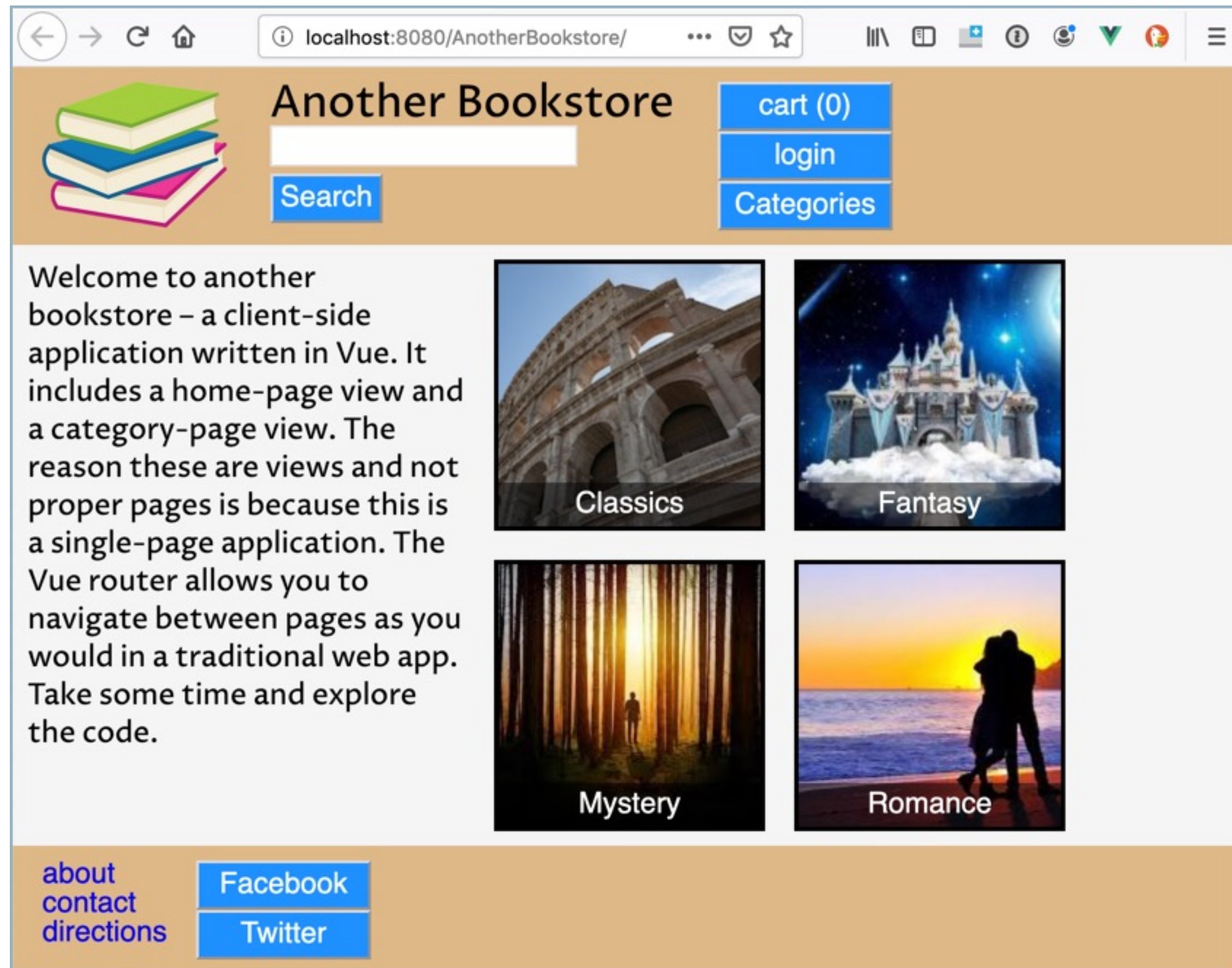
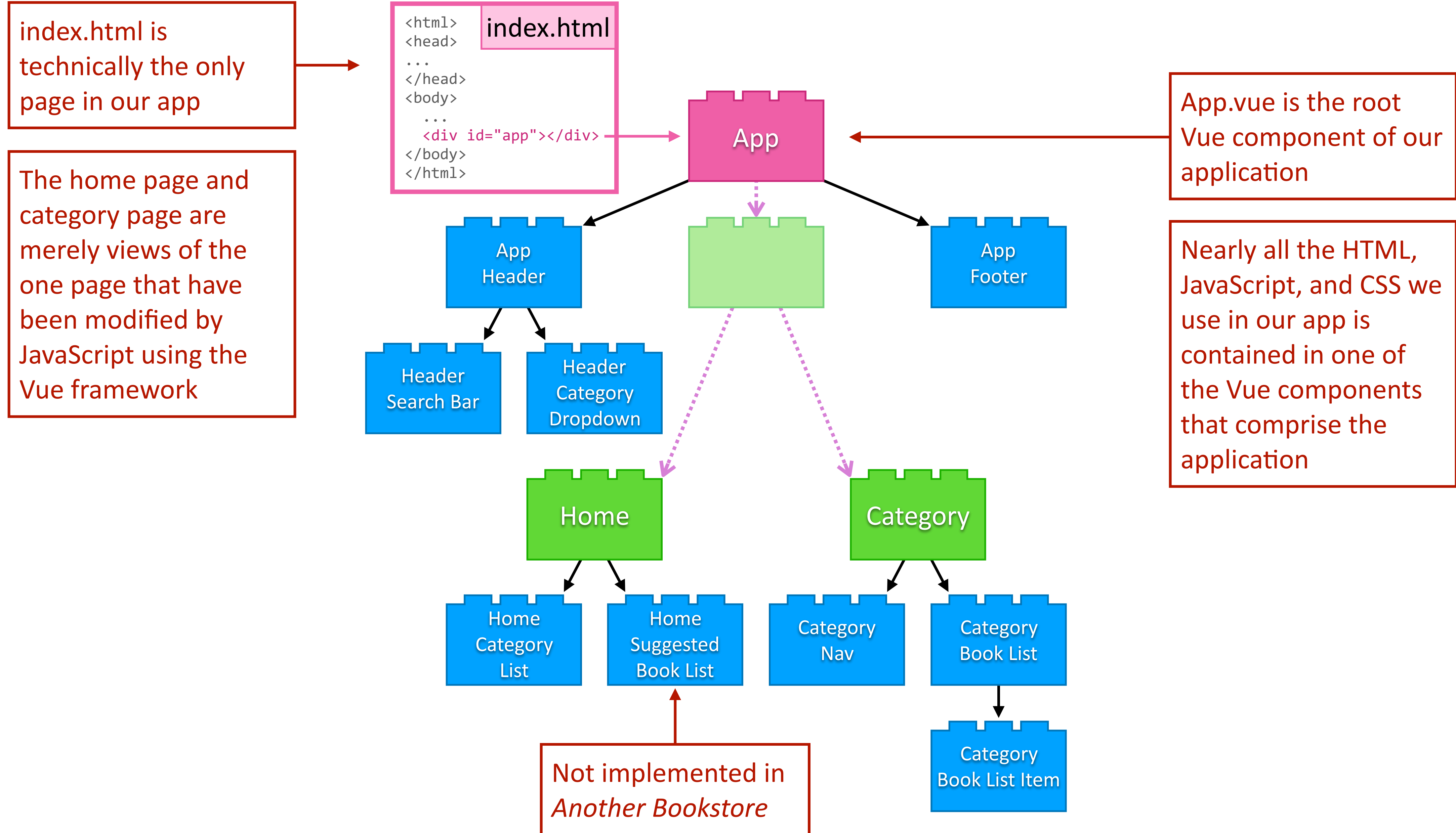


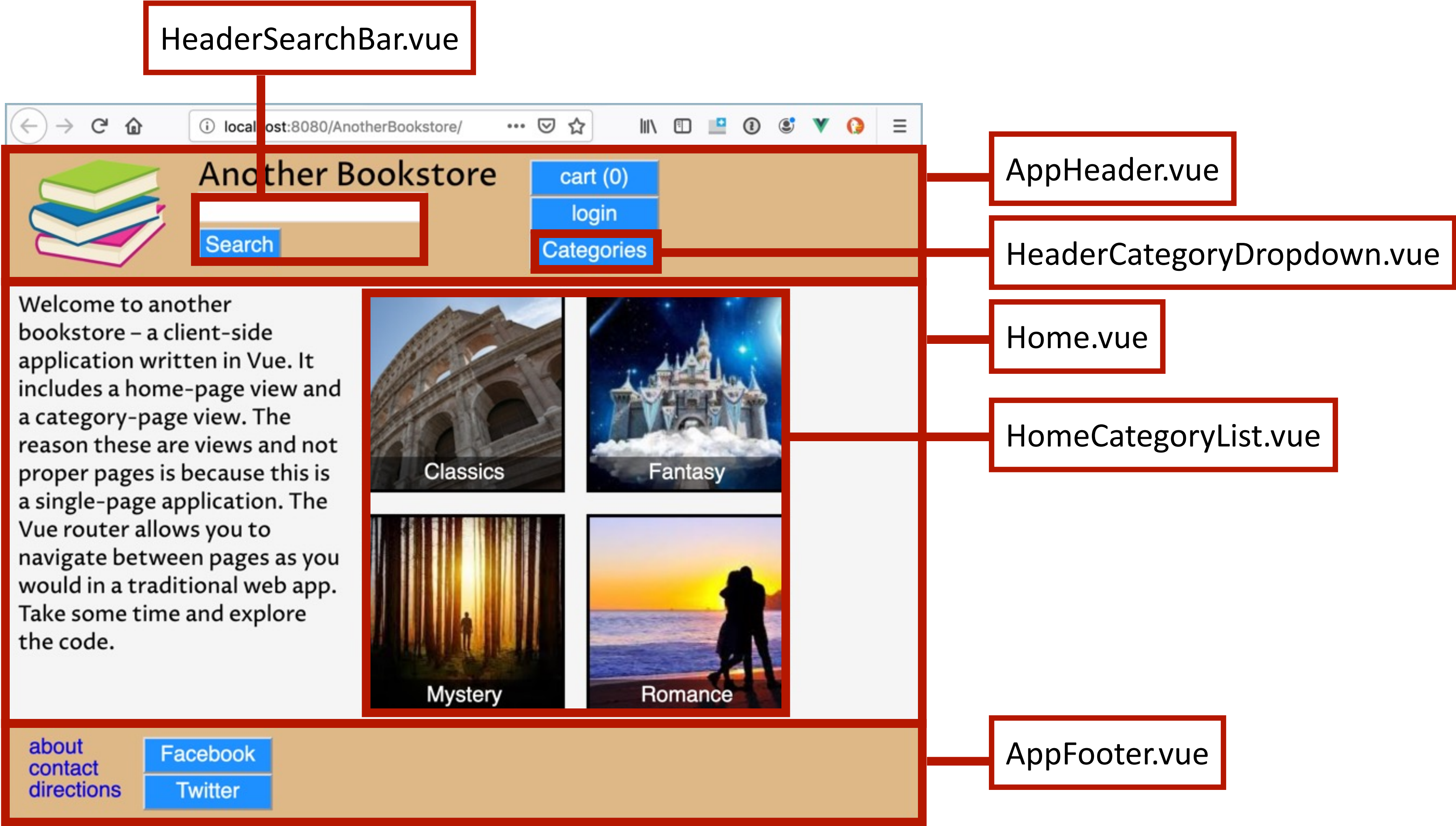
# Another Bookstore







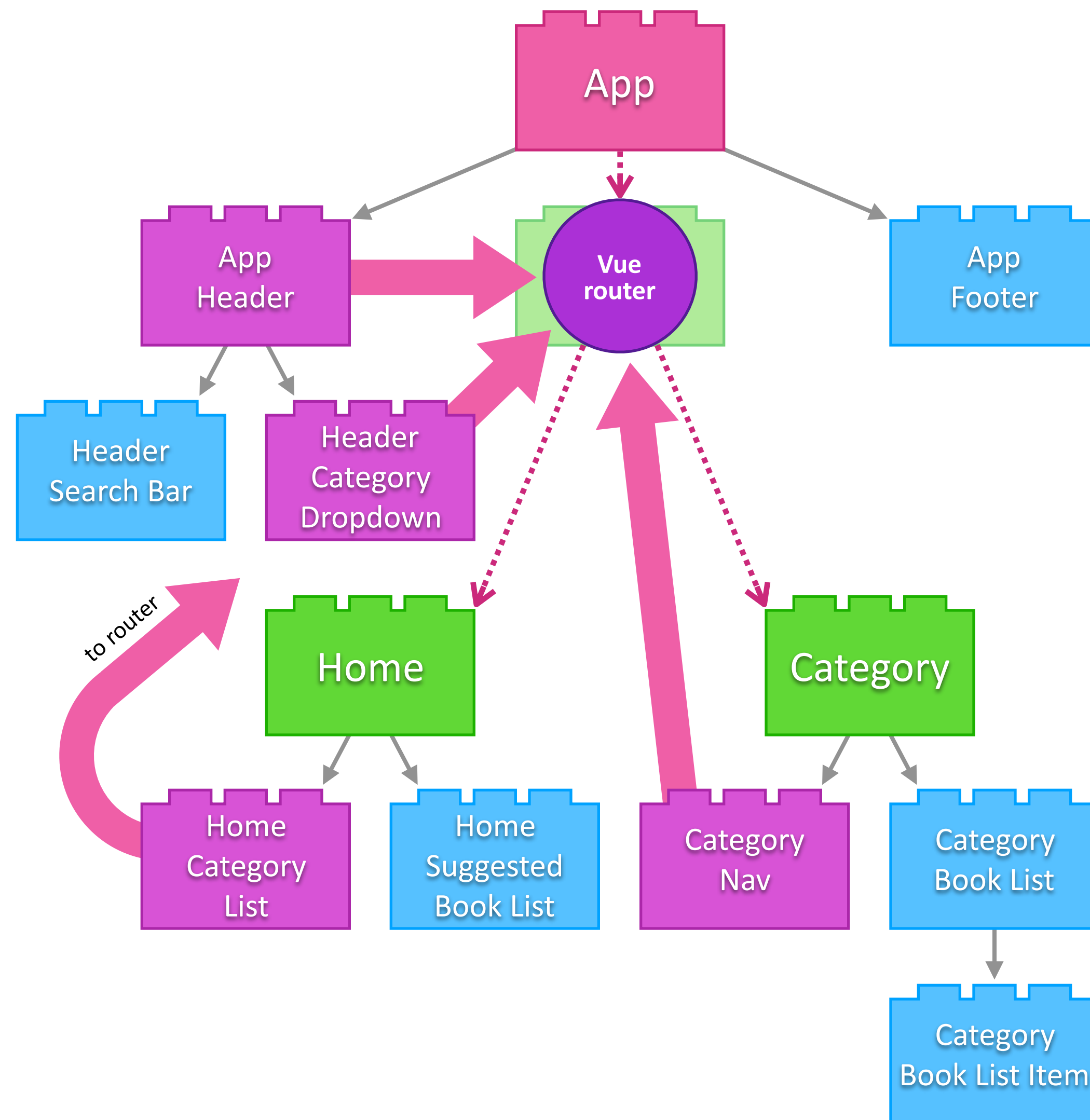
A Vue component contains the HTML, JS, and CSS for a particular part of our application



The Vue router handles page updates for a single page application in Vue

In our app, there are 4 components that can change the route. They use `<router-link>` tags with the `to="..."` attribute, similar to `<a>` tags with the `href="..."` attribute in traditional HTML

A `<router-link>` makes use of an `<a>` link. So if you have style problem with `<router-link>`, try changing the style of `<a>` link






## router.js

```
import Vue from "vue";
import Router from "vue-router";
import Home from "@views/Home.vue";
import Category from "@views/Category.vue"


Vue.use(Router);

export default new Router({
  mode: "history",
  base: process.env.BASE_URL,
  routes: [
    {
      path: "/",
      name: "home",
      component: Home
    },
    {
      path: "/category/:name",
      name: "category",
      component: Category,
      props: true
    }
  ]
});
```

When a *router-link* uses a path of "/", the router will re-assemble the page using the Home component in *router-view*



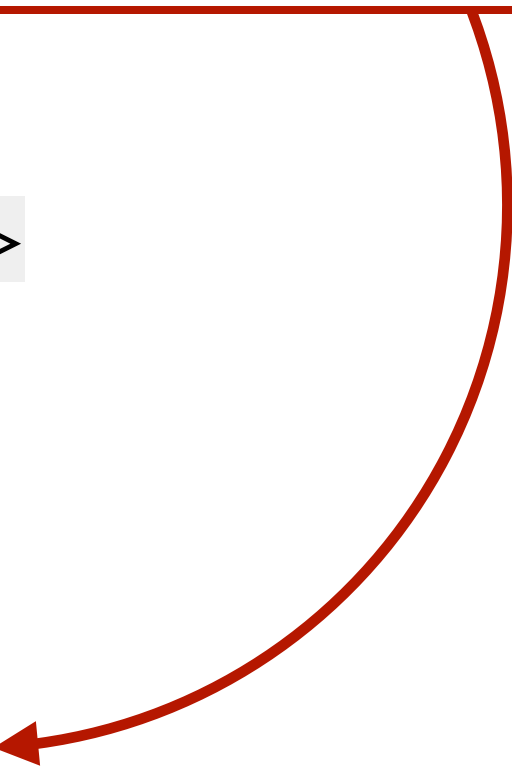
A path of "/category/mystery" will make the router use the Category component, and "mystery" will be the value of *\$route.params.name*



## HeaderCategoryDropdown.vue

```
<template>
  <div id="categoryDropdown">
    <button>Categories</button>
    <ul>
      <li>
        <router-link to=" ../category/classics">Classics</router-link>
      </li>
      <li>
        <router-link to=" ../category/fantasy">Fantasy</router-link>
      </li>
      <li>
        <router-link to=" ../category/mystery">Mystery</router-link>
      </li>
      <li>
        <router-link to=" ../category/romance">Romance</router-link>
      </li>
    </ul>
  </div>
</template>
```

A path of `"/category/mystery"` will make the router use the Category component, and `"mystery"` will be the value of `$route.params.name`



## App.vue

```
<template>
  <div id="app">
    <app-header></app-header>
    <router-view></router-view>
    <app-footer></app-footer>
  </div>
</template>

<script>
import AppHeader from "@components/AppHeader";
import AppFooter from "@components/AppFooter";

export default {
  name: "App",
  components: {
    AppHeader,
    AppFooter
  }
};
</script>
```

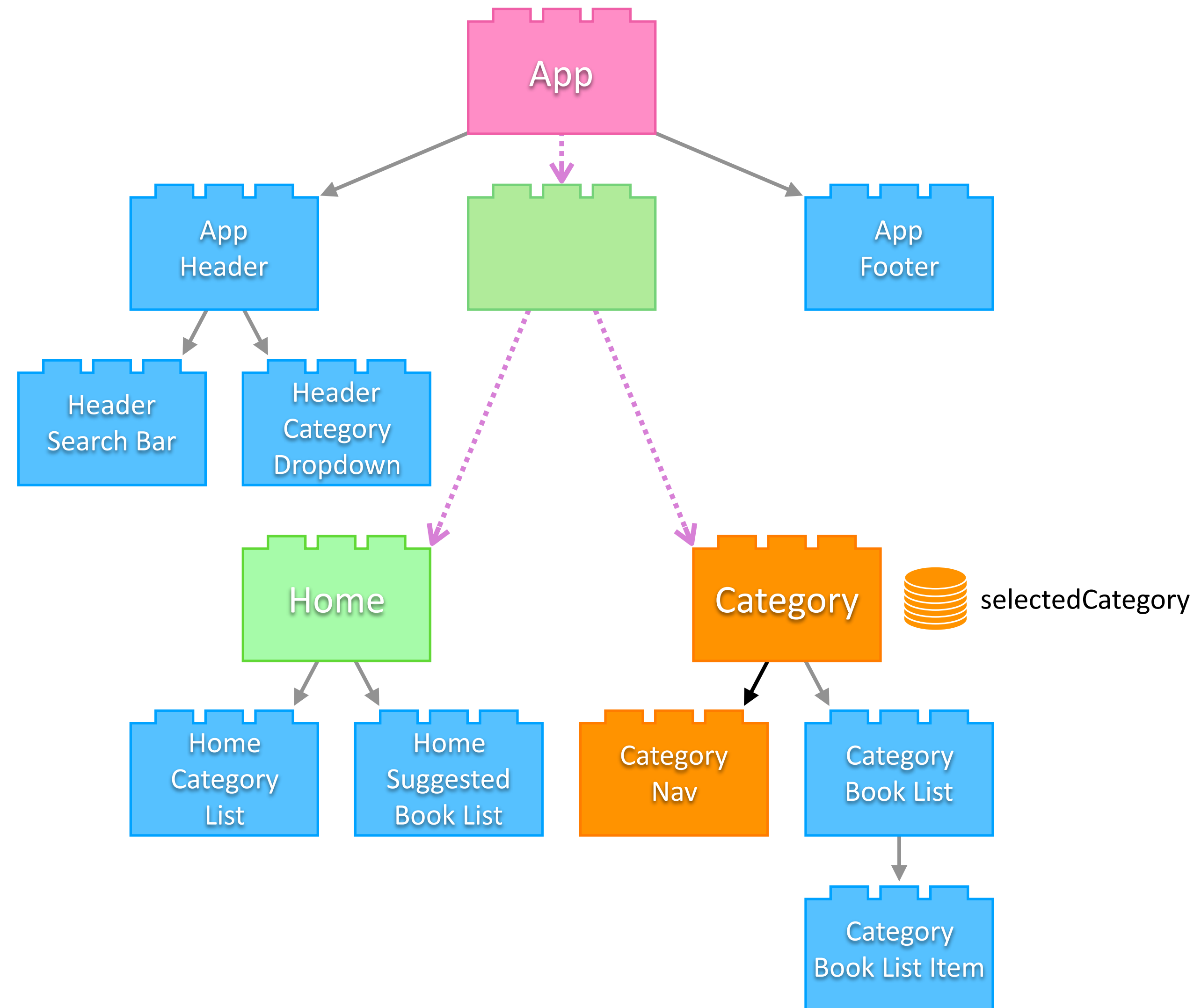
Here is where the router will  
insert the appropriate view  
component



The selected category can be obtained by the Category component using the dynamic-route-matching parameter *\$route.params.name*

The selected category needs to be passed to CategoryNav so we can highlight the selected category button

The current version of the app is not dynamic, so we don't do this, but future versions will

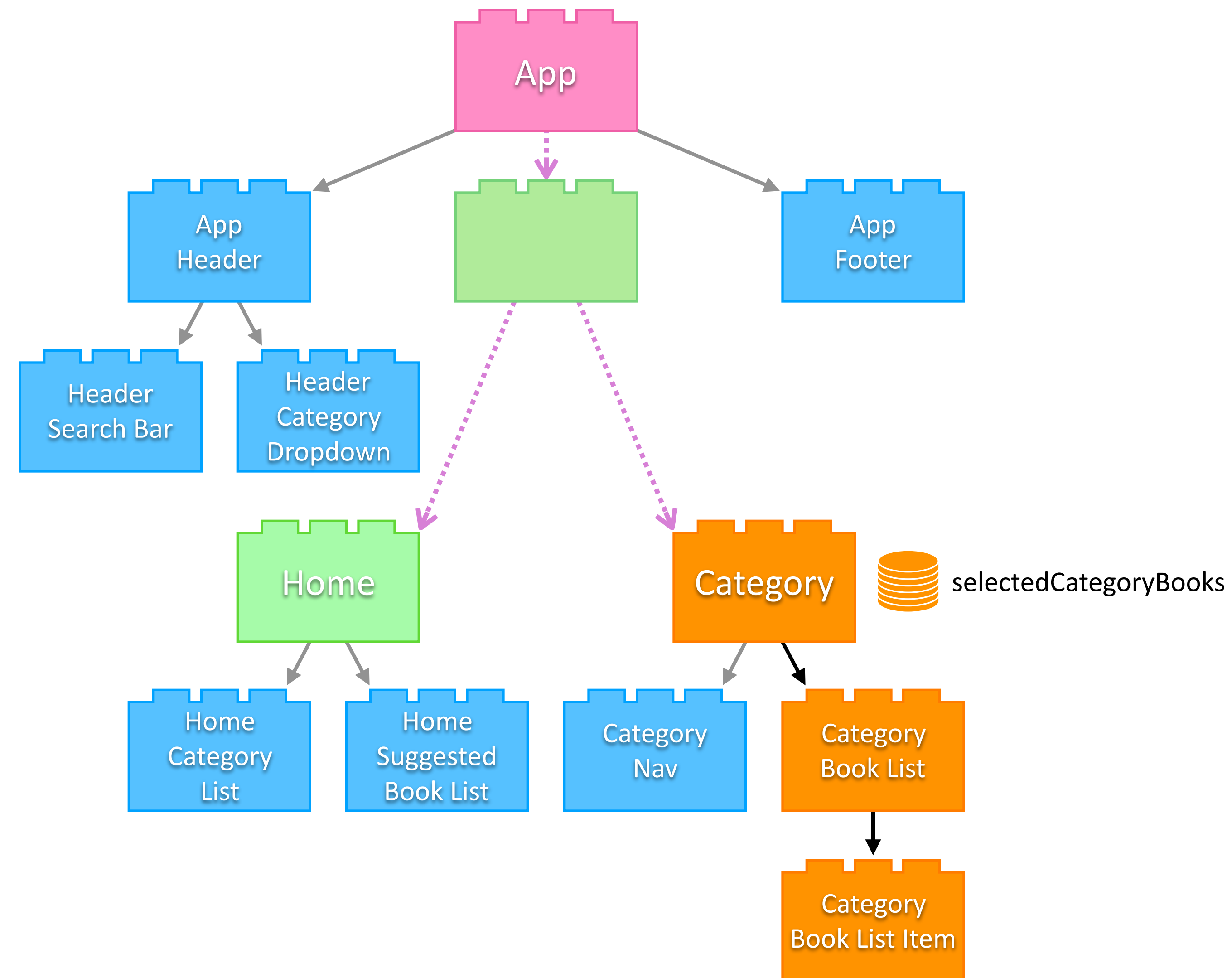




The selected category is also needed to get the list of books associated with that category

The books will ultimately be obtained from a REST service, but for now, the list is given as an array of JSON objects

The array is passed to the CategoryBookList as a prop



## Category.vue

```
<template>
  <div>
    <category-nav></category-nav>
    <category-book-list :books="books"> </category-book-list>
  </div>
</template>
```

The array of books is passed to the CategoryBookList component

```
<script>
import CategoryNav from "@components/CategoryNav";
import CategoryBookList from "@components/CategoryBookList";
```

```
export default {
  name: "Category",
  components: {
    CategoryNav,
    CategoryBookList
  },
  data: function() {
    return {
      books: [
        {
          bookId: 1001,
          title: "The Iliad",
          imageFile: "the-iliad.gif",
          author: "Homer",
          price: 699,
          isPublic: true
        },
        ...
      ]
    }
  }
}
```

These are the two components used by the Category component. They need to be imported and listed in the components field

In Vue components (unlike in Vue objects) the data field is a function

Here it is returning an array of book objects

CategoryBookList.vue

```
<template>
  <ul>
    <category-book-list-item :book="books[0]"></category-book-list-item>
    <category-book-list-item :book="books[1]"></category-book-list-item>
    <category-book-list-item :book="books[2]"></category-book-list-item>
    <category-book-list-item :book="books[3]"></category-book-list-item>
  </ul>
</template>
```

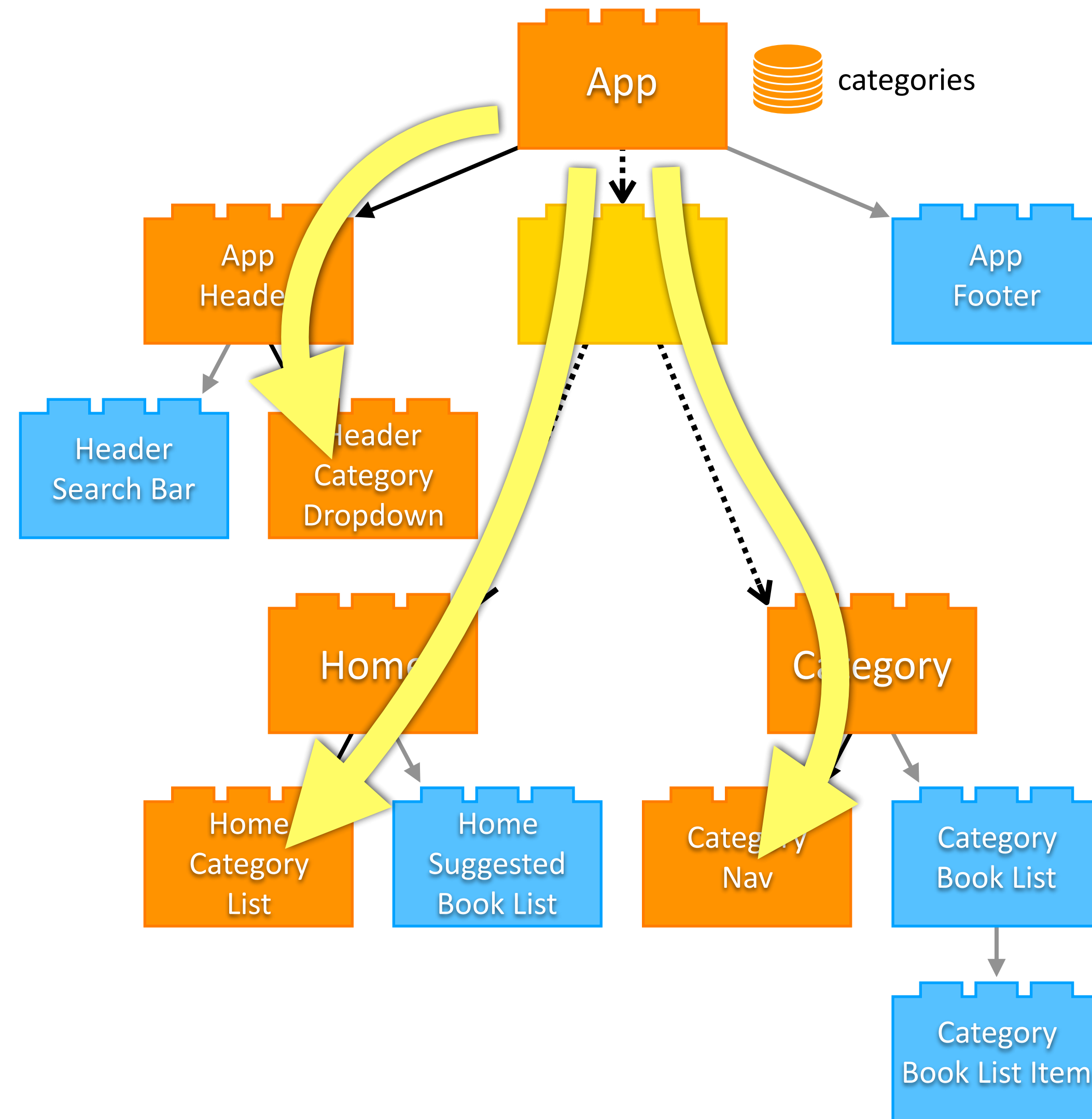
```
<script>
import CategoryBookListItem from "@components/CategoryBookListItem";

export default {
  name: "CategoryBookList",
  props: {
    books: {
      type: Array,
      required: true
    }
  },
  components: {
    CategoryBookListItem
  }
};
</script>
```

Each book object from the array is passed to a CategoryBookListItem component

Arguments taken are listed in the props field





Take a look at what we would have to do to have the categories calculated in one place and passed down to all the components that needed them

Since there are only 4 categories, the current version of our app simply hard-codes them into the 3 components that need them, but this is redundant

In a later project, we will make the categories globally available, so they are neither passed as props nor duplicated across components