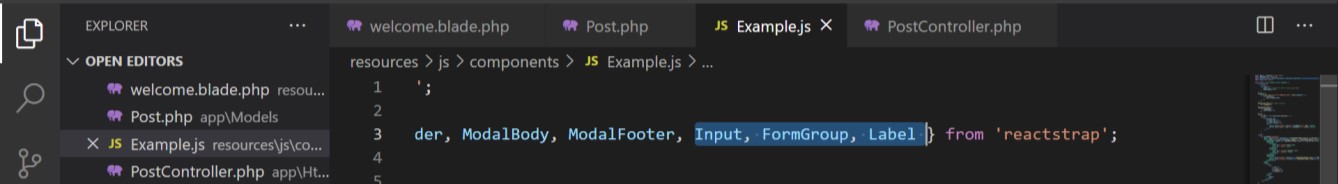
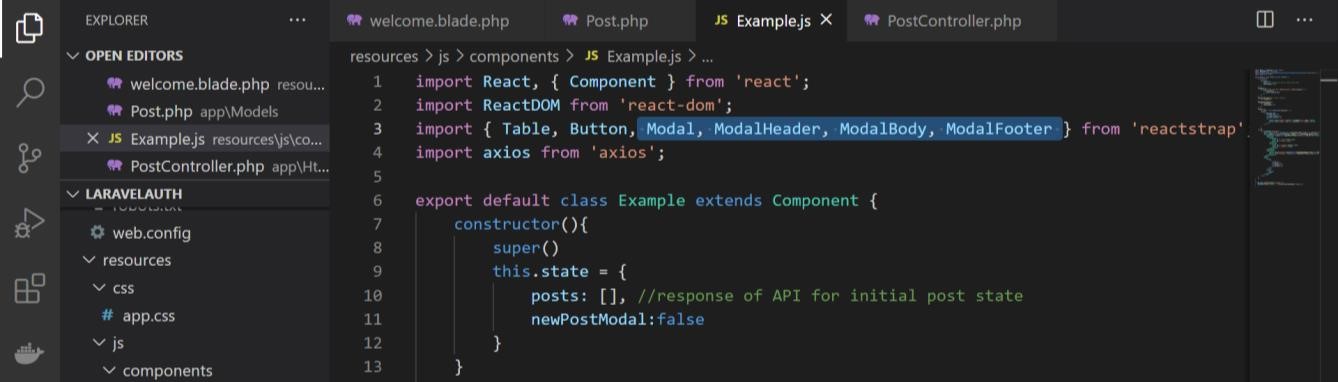
**Practical 8 : Laravel ReactJS CRUD with RESTful API (Part II).**

In this lab, we will continue to explore the usage of client-side scripts using a JavaScript library. In this case, we will explore ReactJS library with Laravel with the continuation of previous laravelAuth web application that we have created for Laravel Authentication and Authorization.

**1. Define Front End for Adding, Updating and Deleting a Post.**

We can choose to use Modals in reactstrap to pop up pages so that user may interact the different CRUD through pop up forms (modals). Import components of Modal and declare default state of Modal as shown in Figure 1.



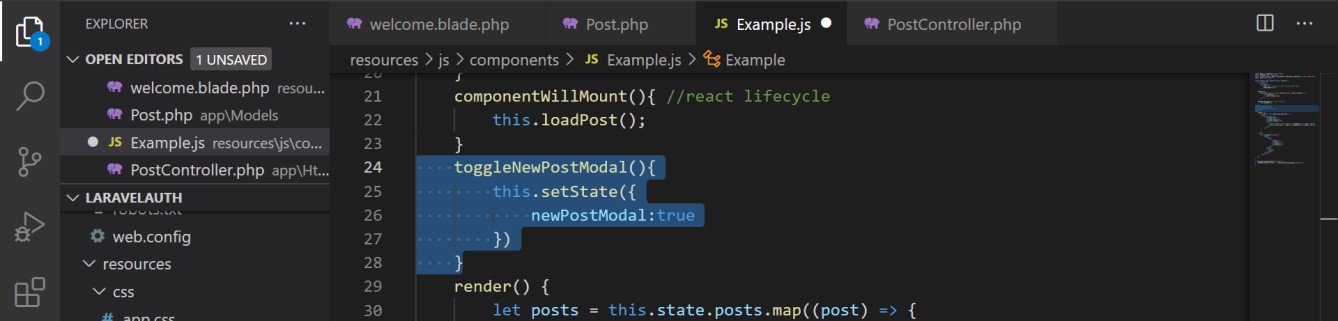


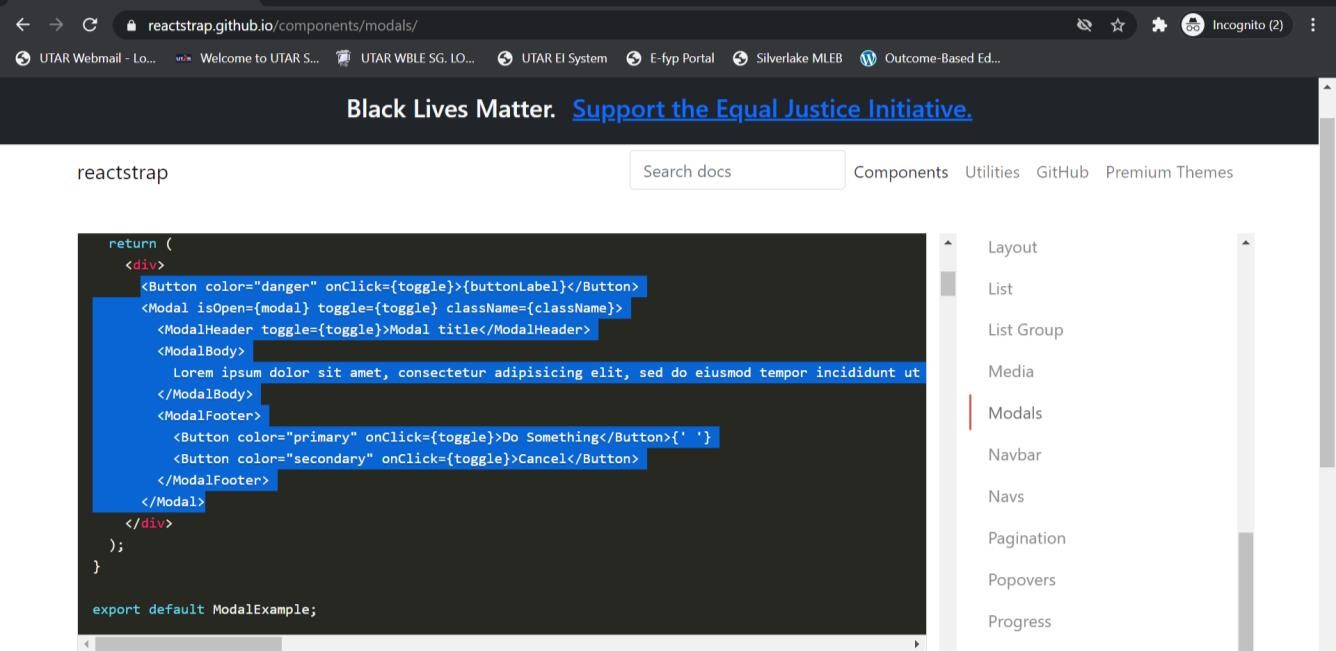
Figure 1: Initialize Modal and Form libraries and Modal’s default state.

Define a state change for modal when a trigger (button onClick) is executed as shown in Figure 2. Figure 2: Change of Modal state.

Define the Modal through example codes provided in React Bootstrap 4

[(https://reactstrap.github.io/)](https://reactstrap.github.io/) documentation as shown in Figure 3.

Figure 3: Modal example code.



Modify the code as shown in Figure 4 to create binding actions; bind the ‘onClick’ action of the button to set the Modal’s state. This will complete the modal for adding a new post.

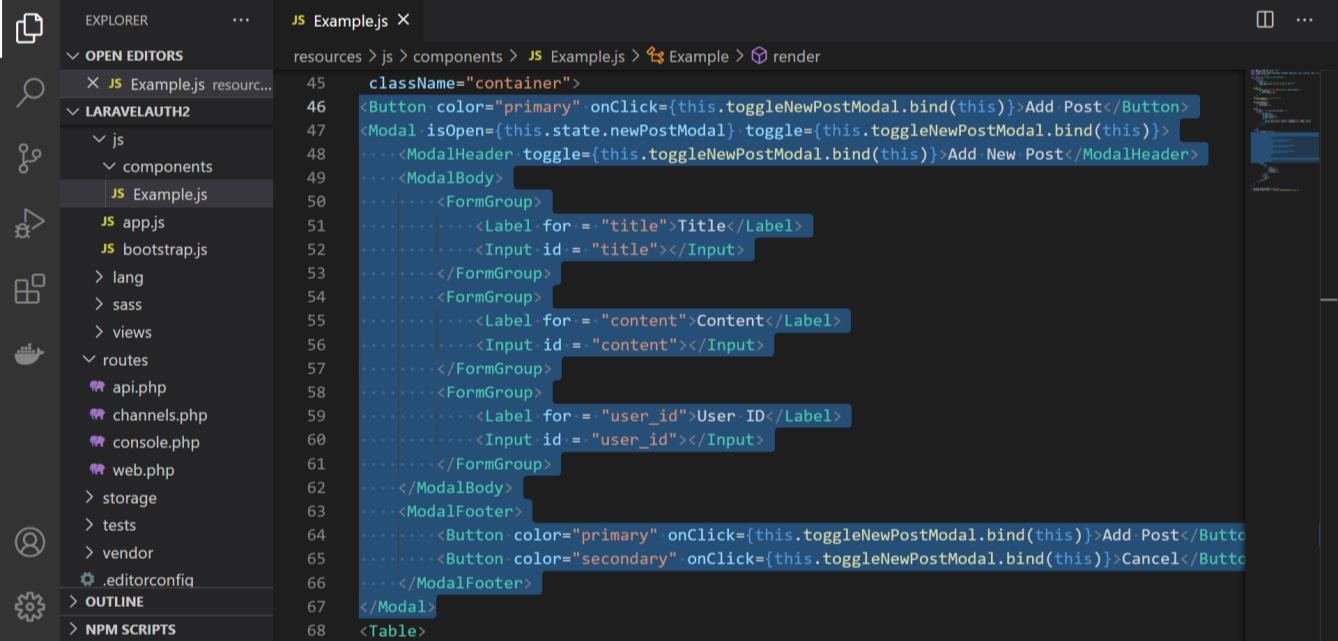


Figure 4: Define Modal for Add Post.

**<Button color="primary" onClick={this.toggleNewPostModal.bind(this)}>Add**

**Post</Button>**

**<Modal isOpen={this.state.newPostModal}**

**toggle={this.toggleNewPostModal.bind(this)}>**

**<ModalHeader toggle={this.toggleNewPostModal.bind(this)}> Add New Post**

**</ModalHeader>**

**<ModalBody>**

**<FormGroup>**

**<Label for = "title">Title</Label>**

**<Input id = "title"></Input>**

**</FormGroup>**

**<FormGroup>**

**<Label for = "content">Content</Label>**

**<Input id = "content"></Input>**

**</FormGroup>**

**<FormGroup>**

**<Label for = "user\_id">User ID</Label>**

**<Input id = "user\_id"></Input>**

**</FormGroup>**

**</ModalBody>**

**<ModalFooter>**

**<Button color="primary" onClick={this.toggleNewPostModal.bind(this)}>**

**Add Post </Button>{' '}**

**<Button color="secondary"**

**onClick={this.toggleNewPostModal.bind(this)}> Cancel </Button>**

**</ModalFooter>**

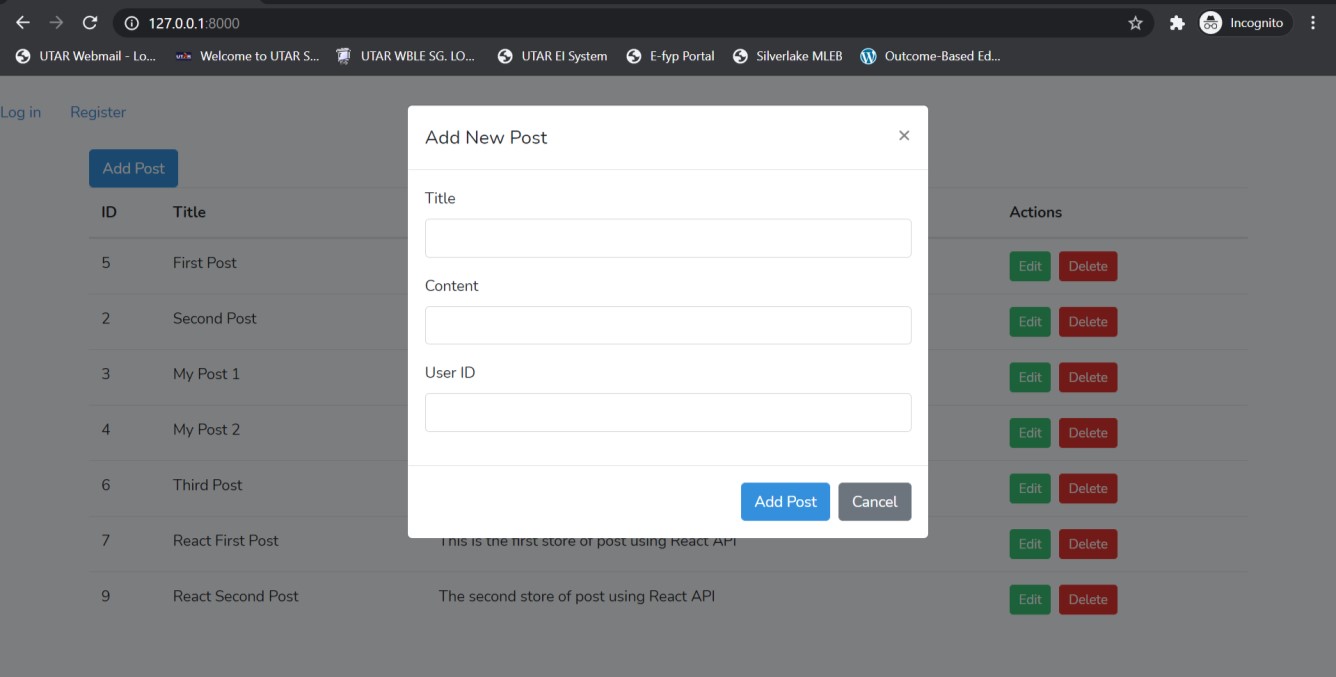
**</Modal>**

Compile JS and CSS assets and host the web application to see the Modal pop up prior to onClick event of “Add Post” button as shown in Figure 5.

Figure 5:

**newPostModal: true**

.



|  |
| --- |
| Example4.js  import React, { Component } from 'react';  import ReactDOM from 'react-dom';  import { Table, Button, Modal, ModalHeader, ModalBody, ModalFooter, FormGroup, Label, Input } from 'reactstrap';  import axios from 'axios';  export default class Example extends Component {      constructor() {          super()          this.state = {              posts: [], //response of API into post state              newPostModal: false,              newPostData: { title: "", content: "", user\_id: "" }          }      }      loadPost() {          axios.get('http://127.0.0.1:8000/api/posts').then((response) => {              this.setState({                  posts: response.data              })          })      }      addPost() {          axios.post('http://127.0.0.1:8000/api/post', this.state.newPostData).then((response) => {              let { posts } = this.state              this.loadPost()              this.setState({                  posts,                  newPostModal: false,                  newPostData: { title: "", content: "", user\_id: "" }              })          })      }      componentWillMount() {          this.loadPost();      }      toggleNewPostModal() {          this.setState({              newPostModal: true          })      }      render() {          let posts = this.state.posts.map((post) => {              return (                  <tr key={post.id}>                      <td>{post.id}</td>                      <td>{post.title}</td>                      <td>{post.content}</td>                      <td>                          <Button color="success" size="sm" className="mr-2"> Edit </Button>                          <Button color="danger" size="sm" className="mr-2"> Delete </Button>                      </td>                  </tr>              )          })          return (              <div className="container">                  <Button color="primary" onClick={this.toggleNewPostModal.bind(this)}>Add Post</Button>                  <Modal isOpen={this.state.newPostModal} toggle={this.toggleNewPostModal.bind(this)}>                      <ModalHeader toggle={this.toggleNewPostModal.bind(this)}> Add New Post                      </ModalHeader>                      <ModalBody>                          <FormGroup>                              <Label for="title">Title</Label>                              <Input                                  id="title"                                  value={this.state.newPostData.title}                                  onChange={(e) => {                                      let { newPostData } = this.state                                      newPostData.title = e.target.value                                      this.setState({ newPostData })                                  }}                              ></Input>                          </FormGroup>                          <FormGroup>                              <Label for="content">Content</Label>                              <Input                                  id="content"                                  value={this.state.newPostData.content}                                  onChange={(e) => {                                      let { newPostData } = this.state                                      newPostData.content = e.target.value                                      this.setState({ newPostData })                                  }}                              ></Input>                          </FormGroup>                          <FormGroup>                              <Label for="user\_id">User ID</Label>                              <Input                                  id="user\_id"                                  value={this.state.newPostData.user\_id}                                  onChange={(e) => {                                      let { newPostData } = this.state                                      newPostData.user\_id = e.target.value                                      this.setState({ newPostData })                                  }}                              ></Input>                          </FormGroup>                      </ModalBody>                      <ModalFooter>                          <Button color="primary" onClick={this.addPost.bind(this)}> Add Post </Button>{' '}                          <Button color="secondary" onClick={this.toggleNewPostModal.bind(this)}> Cancel </Button>                      </ModalFooter>                  </Modal>                  <Table>                      <thead>                          <tr>                              <th>ID</th>                              <th>Title</th>                              <th>Content</th>                              <th>Actions</th>                          </tr>                      </thead>                      <tbody>                          {posts}                      </tbody>                  </Table>              </div>          );      }  }  if (document.getElementById('example')) {      ReactDOM.render(<Example />, document.getElementById('example'));  } |

The Modal has no further actions for all of the buttons on the Modal, for there are no definitions of setting back the state to be FALSE after the state was set to be TRUE prior to the onClick of “Add Post” button.

Since this modal is for the purpose of adding a new post, let’s create a method/function for “post” API as shown in Figure 6.

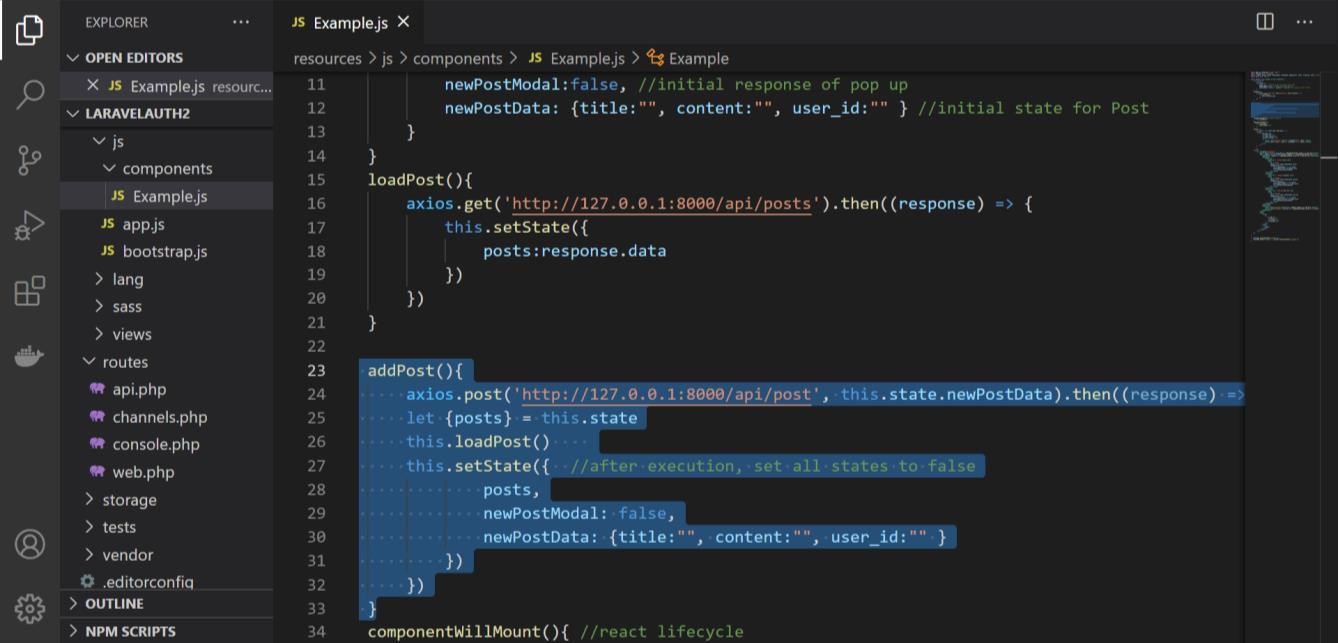


Figure 6: addPost method/function to add a new post.

Prior to creation of addPost method/function, modify the Add Post modal’s form components in order to send the request from user to API for processing of adding new post as shown in Figure 7.

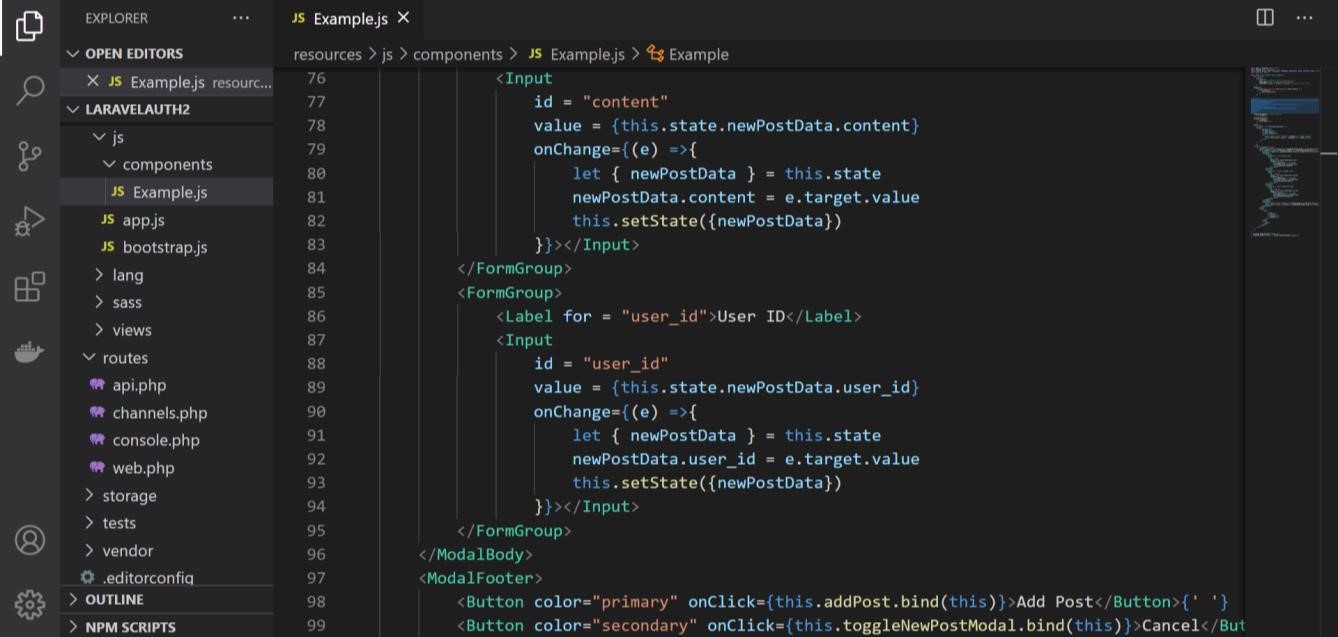
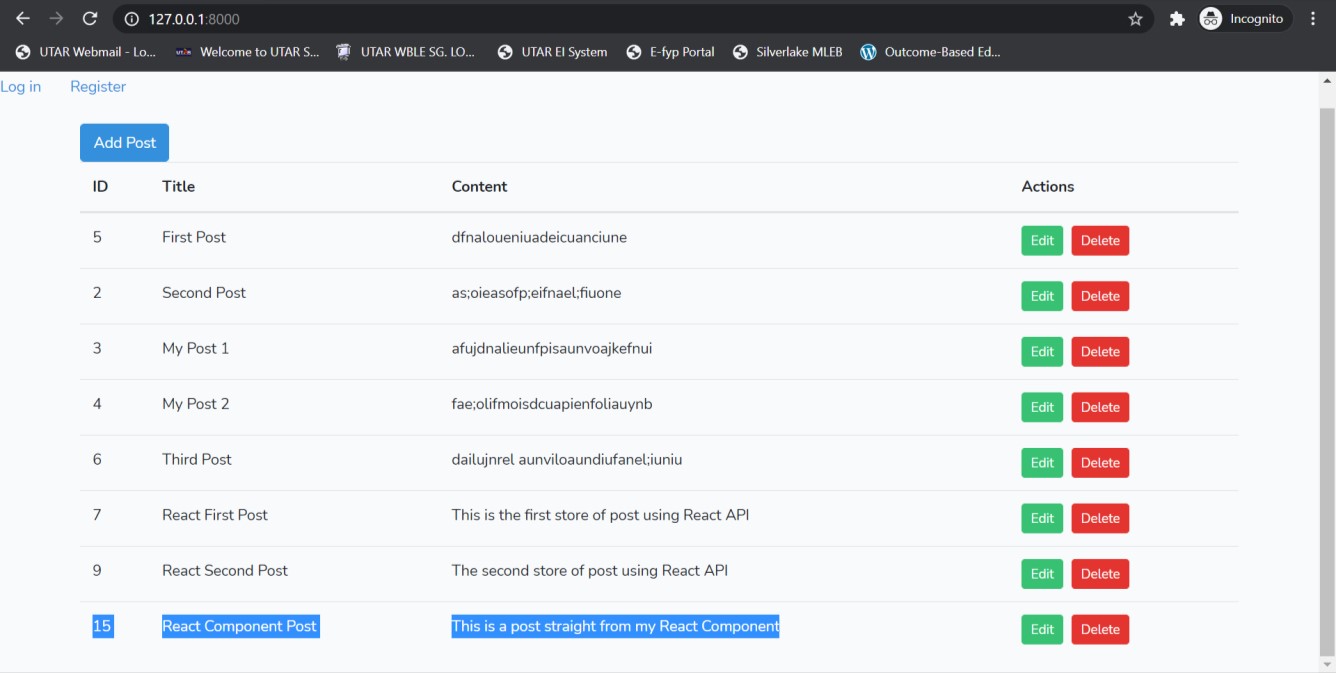


Figure 7: setStates to fetch the value from HTTP request (user inputs).

Then, compile JS and CSS assets and refresh the web application to create a new post as shown in Figure 8.

Figure 8: Add new post using React Component.



------------------------------------------------------------------------------------------------------------------------

Now, let’s proceed to create edit/update method/function to fetch the data from the table and another method/function for the “put” API in React Component to update Post as shown in Figure 9.

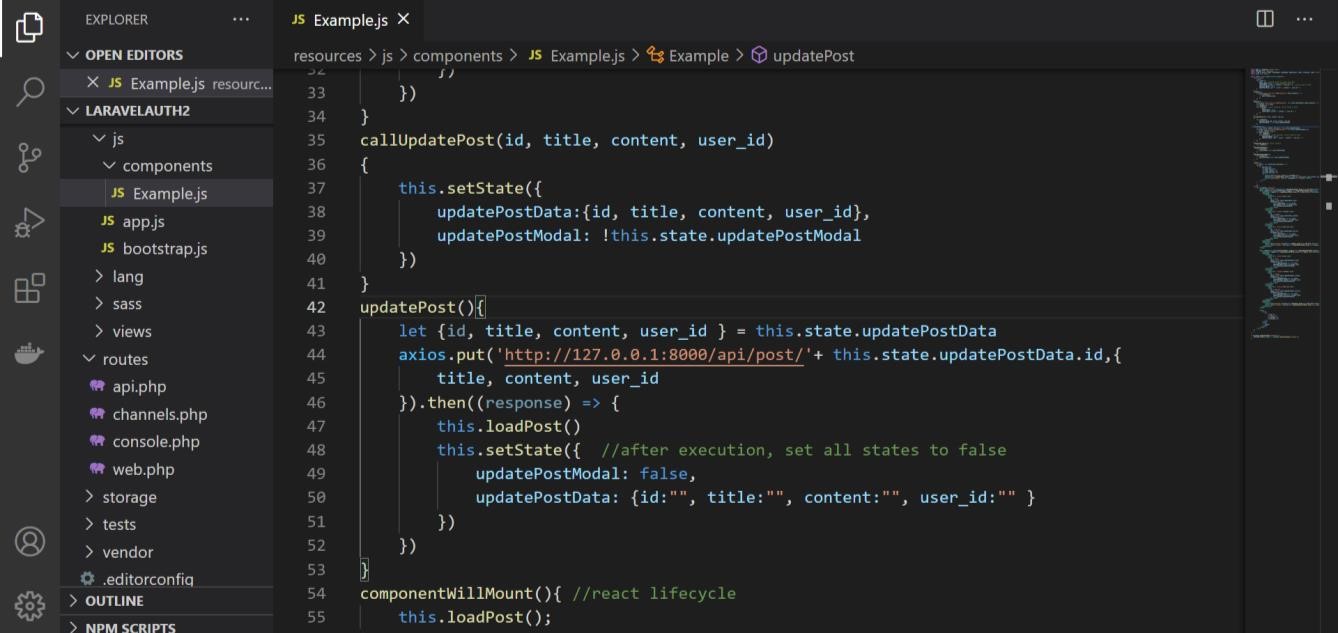


Figure 9: A method for fetching and changing modal state and a method for “put” API.

|  |
| --- |
| **callUpdatePost(id, title, content, user\_id)**  **{**  **this.setState({**  **updatePostData:{id, title, content, user\_id}, updatePostModal: !this.state.updatePostModal**  **}) }**  **updatePost(){**  **let {id, title, content, user\_id } = this.state.updatePostData**  **axios.put('http://127.0.0.1:8000/api/post/'+ this.state.updatePostData.id,{ title, content, user\_id }).then((response) => { this.loadPost()**  **this.setState({ //after execution, set all states to false updatePostModal: false,**  **updatePostData: {id:"", title:"", content:"", user\_id:"" }**  **})**  **})**  **} toggleNewPostModal(){ this.setState({**  **newPostModal:!this.state.newPostModal**  **}) } toggleUpdatePostModal(){ this.setState({**  **updatePostModal:!this.state.updatePostModal**  **})**  **}** |

Then, bind the “Edit” buttons in Actions to callUpdatePost method/function as shown in Figure 10.

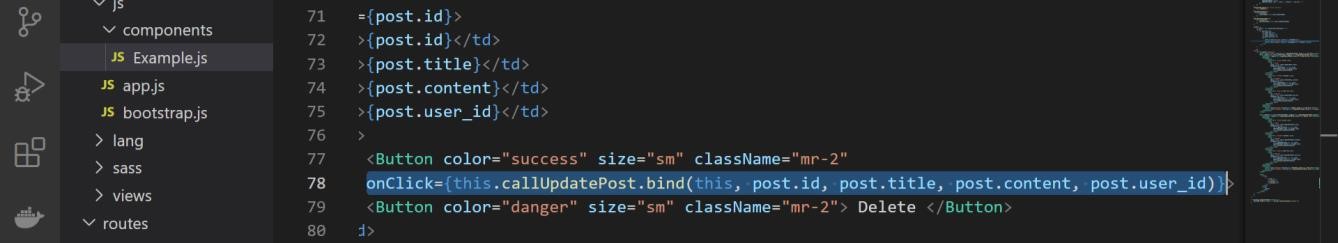


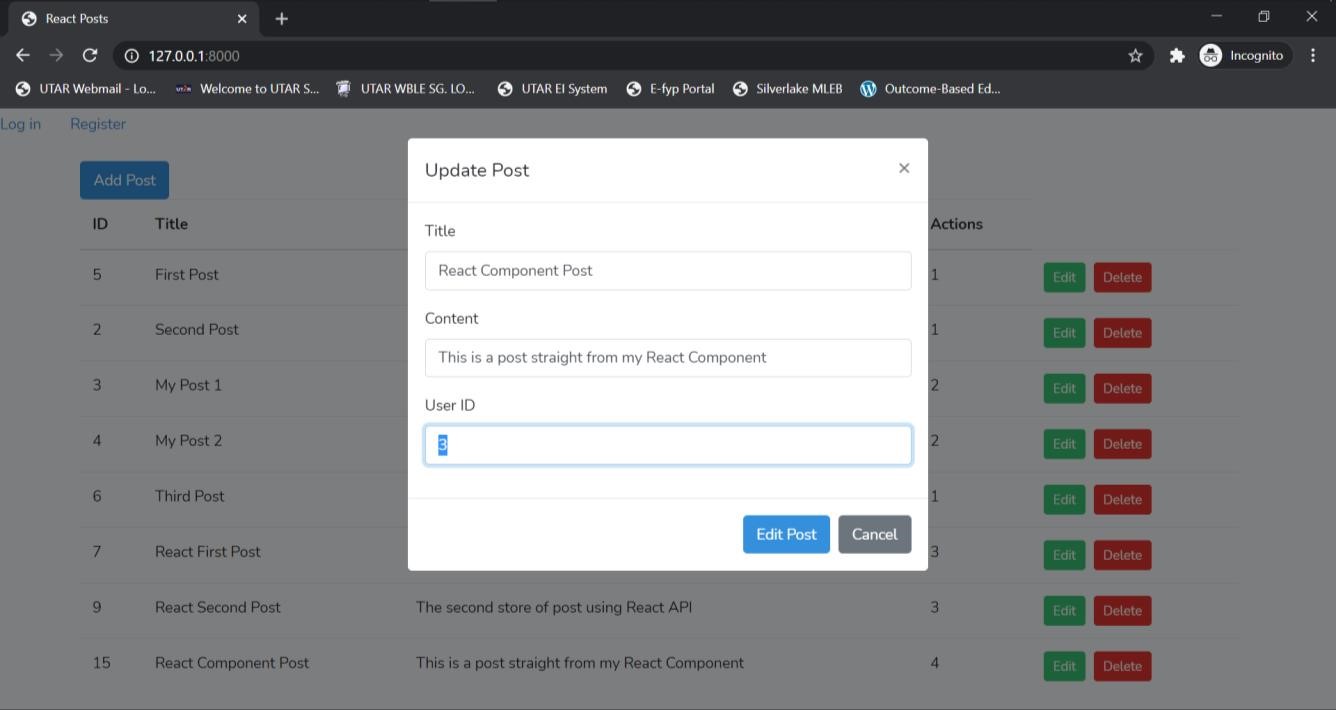
Figure 10: Bind callUpdatePost() to Edit button to invoke modal and API.

**Exercise.**

Create updatePostModal in order to be pop up, once onClick of Edit Button is triggered and bind the modal to toggleUpdatePostModal (*Hint: similar to what was done in newPostModal*).

Subsequent to creation of updatePostModal, proceed to test updating a post using React Component as shown in Figure 11.

Figure 11: Update post using React Component.

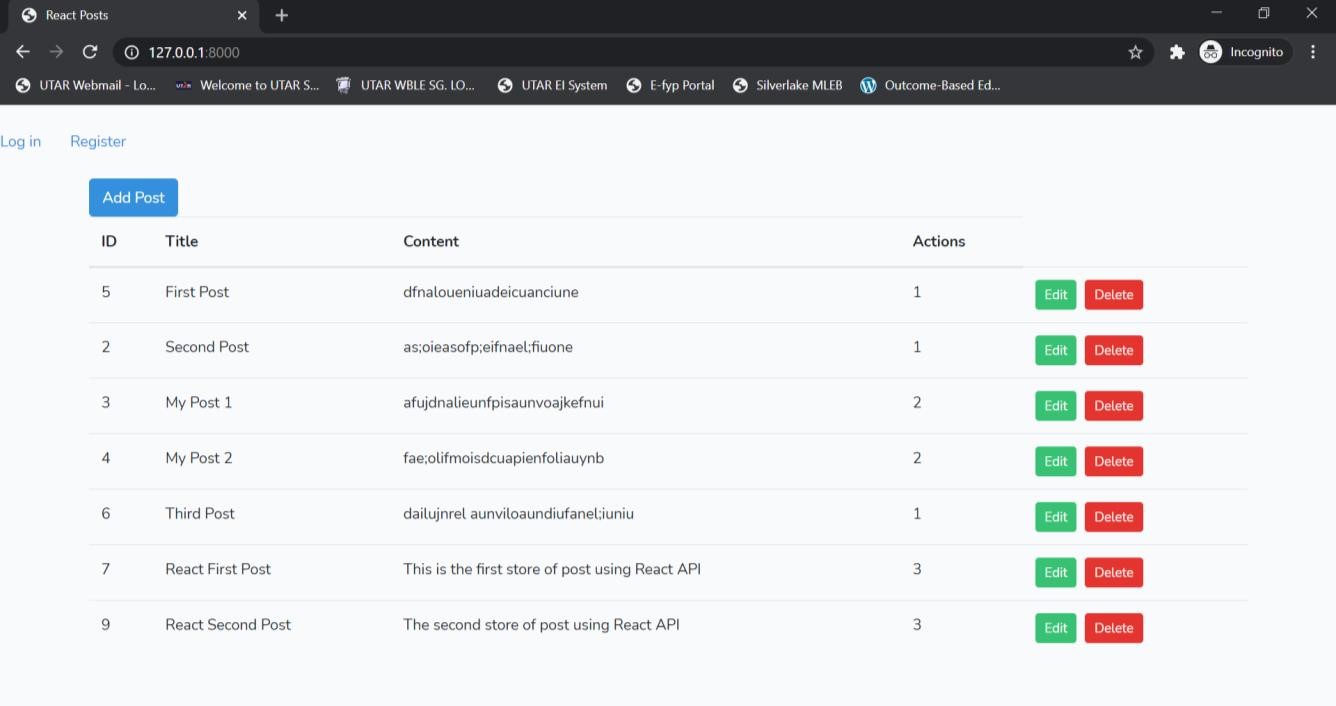


|  |
| --- |
| Example5.js  import React, { Component } from 'react';  import ReactDOM from 'react-dom';  import { Table, Button, Modal, ModalHeader, ModalBody, ModalFooter, FormGroup, Label, Input } from 'reactstrap';  import axios from 'axios';  export default class Example extends Component {      constructor() {          super()          this.state = {              posts: [], //response of API into post state              newPostModal: false,              updatePostModal: false,              newPostData: { title: "", content: "", user\_id: "" },              updatePostData: { id: "", title: "", content: "", user\_id: "" }          }      }      loadPost() {          axios.get('http://127.0.0.1:8000/api/posts').then((response) => {              this.setState({                  posts: response.data              })          })      }      addPost() {          axios.post('http://127.0.0.1:8000/api/post', this.state.newPostData).then((response) => {              let { posts } = this.state              this.loadPost()              this.setState({                  posts,                  newPostModal: false,                  newPostData: { title: "", content: "", user\_id: "" },              })          })      }      componentWillMount() {          this.loadPost();      }      toggleNewPostModal() {          this.setState({ newPostModal: !this.state.newPostModal })      }      callUpdatePost(id, title, content, user\_id) {          this.setState({              updatePostData: { id, title, content, user\_id },              updatePostModal: !this.state.updatePostModal          })      }      updatePost() {          let { id, title, content, user\_id } = this.state.updatePostData          axios.put('http://127.0.0.1:8000/api/post/' + this.state.updatePostData.id, {              title, content, user\_id          }).then((response) => {              this.loadPost()              this.setState({ //after execution, set all states to false                  updatePostModal: false,                  updatePostData: { id: "", title: "", content: "", user\_id: "" }              })          })      }      toggleUpdatePostModal() {          this.setState({              updatePostModal: !this.state.updatePostModal          })      }      render() {          let posts = this.state.posts.map((post) => {              return (                  <tr key={post.id}>                      <td>{post.id}</td>                      <td>{post.title}</td>                      <td>{post.content}</td>                      <td>                          <Button color="success" size="sm" className="mr-2"                              onClick={this.callUpdatePost.bind(this, post.id, post.title, post.content, post.user\_id)}>                              Edit </Button>                          <Button color="danger" size="sm" className="mr-2"> Delete </Button>                      </td>                  </tr>              )          })          return (              <div className="container">                  <Button color="primary" onClick={this.toggleNewPostModal.bind(this)}>Add Post</Button>                  <Modal isOpen={this.state.newPostModal} toggle={this.toggleNewPostModal.bind(this)}>                      <ModalHeader toggle={this.toggleNewPostModal.bind(this)}> Add New Post                      </ModalHeader>                      <ModalBody>                          <FormGroup>                              <Label for="title">Title</Label>                              <Input                                  id="title"                                  value={this.state.newPostData.title}                                  onChange={(e) => {                                      let { newPostData } = this.state                                      newPostData.title = e.target.value                                      this.setState({ newPostData })                                  }}                              ></Input>                          </FormGroup>                          <FormGroup>                              <Label for="content">Content</Label>                              <Input                                  id="content"                                  value={this.state.newPostData.content}                                  onChange={(e) => {                                      let { newPostData } = this.state                                      newPostData.content = e.target.value                                      this.setState({ newPostData })                                  }}                              ></Input>                          </FormGroup>                          <FormGroup>                              <Label for="user\_id">User ID</Label>                              <Input                                  id="user\_id"                                  value={this.state.newPostData.user\_id}                                  onChange={(e) => {                                      let { newPostData } = this.state                                      newPostData.user\_id = e.target.value                                      this.setState({ newPostData })                                  }}                              ></Input>                          </FormGroup>                      </ModalBody>                      <ModalFooter>                          <Button color="primary" onClick={this.addPost.bind(this)}> Add Post </Button>{' '}                          <Button color="secondary" onClick={this.toggleNewPostModal.bind(this)}> Cancel </Button>                      </ModalFooter>                  </Modal>                  <Modal isOpen={this.state.updatePostModal} toggle={this.toggleUpdatePostModal.bind(this)}>                      <ModalHeader toggle={this.toggleUpdatePostModal.bind(this)}> Update Post                      </ModalHeader>                      <ModalBody>                          <FormGroup>                              <Label for="title">Title</Label>                              <Input                                  id="title"                                  value={this.state.updatePostData.title}                                  onChange={(e) => {                                      let { updatePostData } = this.state                                      updatePostData.title = e.target.value                                      this.setState({ updatePostData })                                  }}                              ></Input>                          </FormGroup>                          <FormGroup>                              <Label for="content">Content</Label>                              <Input                                  id="content"                                  value={this.state.updatePostData.content}                                  onChange={(e) => {                                      let { updatePostData } = this.state                                      updatePostData.content = e.target.value                                      this.setState({ updatePostData })                                  }}                              ></Input>                          </FormGroup>                          <FormGroup>                              <Label for="user\_id">User ID</Label>                              <Input                                  id="user\_id"                                  value={this.state.updatePostData.user\_id}                                  onChange={(e) => {                                      let { updatePostData } = this.state                                      updatePostData.user\_id = e.target.value                                      this.setState({ updatePostData })                                  }}                              ></Input>                          </FormGroup>                      </ModalBody>                      <ModalFooter>                          <Button color="primary" onClick={this.updatePost.bind(this)}> Edit Post </Button>{' '}                          <Button color="secondary" onClick={this.toggleUpdatePostModal.bind(this)}> Cancel </Button>                      </ModalFooter>                  </Modal>                  <Table>                      <thead>                          <tr>                              <th>ID</th>                              <th>Title</th>                              <th>Content</th>                              <th>Actions</th>                          </tr>                      </thead>                      <tbody>                          {posts}                      </tbody>                  </Table>              </div>          );      }  }  if (document.getElementById('example')) {      ReactDOM.render(<Example />, document.getElementById('example'));  } |

**Exercise:**

Create a delete method/function for Delete button to function. Use the knowledge and understandings that you have for add post and edit post using React Component for this exercise. The output is as illustrated in Figure 12 (newly created Post id 15 is deleted subsequent to onClick of Delete button).

Figure 12: Delete post using React Component.



|  |
| --- |
| Example6.js  import React, { Component } from 'react';  import ReactDOM from 'react-dom';  import { Table, Button, Modal, ModalHeader, ModalBody, ModalFooter, FormGroup, Label, Input } from 'reactstrap';  import axios from 'axios';  export default class Example extends Component {      constructor() {          super()          this.state = {              posts: [], //response of API into post state              newPostModal: false,              updatePostModal: false,              newPostData: { title: "", content: "", user\_id: "" },              updatePostData: { id: "", title: "", content: "", user\_id: "" }          }      }      loadPost() {          axios.get('http://127.0.0.1:8000/api/posts').then((response) => {              this.setState({                  posts: response.data              })          })      }      addPost() {          axios.post('http://127.0.0.1:8000/api/post',              this.state.newPostData).then((response) => {                  let { posts } = this.state                  this.loadPost()                  this.setState({                      posts,                      newPostModal: false,                      newPostData: { title: "", content: "", user\_id: "" },                  })              })      }      componentWillMount() {          this.loadPost();      }      toggleNewPostModal() {          this.setState({ newPostModal: !this.state.newPostModal })      }      callUpdatePost(id, title, content, user\_id) {          this.setState({              updatePostData: { id, title, content, user\_id },              updatePostModal: !this.state.updatePostModal          })      }      updatePost() {          let { id, title, content, user\_id } = this.state.updatePostData          axios.put('http://127.0.0.1:8000/api/post/' + this.state.updatePostData.id, {              title, content, user\_id          }).then((response) => {              this.loadPost()              this.setState({ //after execution, set all states to false                  updatePostModal: false,                  updatePostData: { id: "", title: "", content: "", user\_id: "" }              })          })      }      deletePost(id) {          if (confirm("Do you want delete this Post?")) {              axios.delete('http://127.0.0.1:8000/api/post/' + id, {              }).then((response) => {                  this.loadPost()              })          }      }      toggleUpdatePostModal() {          this.setState({              updatePostModal: !this.state.updatePostModal          })      }      render() {          let posts = this.state.posts.map((post) => {              return (                  <tr key={post.id}>                      <td>{post.id}</td>                      <td>{post.title}</td>                      <td>{post.content}</td>                      <td>                          <Button color="success" size="sm" className="mr-2"                              onClick={this.callUpdatePost.bind(this, post.id, post.title, post.content, post.user\_id)}>                              Edit </Button>                          <Button color="danger" size="sm" className="mr-2" onClick={this.deletePost.bind(this, post.id)}> Delete </Button>                      </td>                  </tr>              )          })          return (              <div className="container">                  <Button color="primary" onClick={this.toggleNewPostModal.bind(this)}>Add Post</Button>                  <Modal isOpen={this.state.newPostModal} toggle={this.toggleNewPostModal.bind(this)}>                      <ModalHeader toggle={this.toggleNewPostModal.bind(this)}> Add New Post                      </ModalHeader>                      <ModalBody>                          <FormGroup>                              <Label for="title">Title</Label>                              <Input                                  id="title"                                  value={this.state.newPostData.title}                                  onChange={(e) => {                                      let { newPostData } = this.state                                      newPostData.title = e.target.value                                      this.setState({ newPostData })                                  }}                              ></Input>                          </FormGroup>                          <FormGroup>                              <Label for="content">Content</Label>                              <Input                                  id="content"                                  value={this.state.newPostData.content}                                  onChange={(e) => {                                      let { newPostData } = this.state                                      newPostData.content = e.target.value                                      this.setState({ newPostData })                                  }}                              ></Input>                          </FormGroup>                          <FormGroup>                              <Label for="user\_id">User ID</Label>                              <Input                                  id="user\_id"                                  value={this.state.newPostData.user\_id}                                  onChange={(e) => {                                      let { newPostData } = this.state                                      newPostData.user\_id = e.target.value                                      this.setState({ newPostData })                                  }}                              ></Input>                          </FormGroup>                      </ModalBody>                      <ModalFooter>                          <Button color="primary" onClick={this.addPost.bind(this)}> Add Post </Button>{' '}                          <Button color="secondary" onClick={this.toggleNewPostModal.bind(this)}> Cancel </Button>                      </ModalFooter>                  </Modal>                  <Modal isOpen={this.state.updatePostModal} toggle={this.toggleUpdatePostModal.bind(this)}>                      <ModalHeader toggle={this.toggleUpdatePostModal.bind(this)}>Upadte Post                      </ModalHeader>                      <ModalBody>                          <FormGroup>                              <Label for="title">Title</Label>                              <Input                                  id="title"                                  value={this.state.updatePostData.title}                                  onChange={(e) => {                                      let { updatePostData } = this.state                                      updatePostData.title = e.target.value                                      this.setState({ updatePostData })                                  }}                              ></Input>                          </FormGroup>                          <FormGroup>                              <Label for="content">Content</Label>                              <Input                                  id="content"                                  value={this.state.updatePostData.content}                                  onChange={(e) => {                                      let { updatePostData } = this.state                                      updatePostData.content = e.target.value                                      this.setState({ updatePostData })                                  }}                              ></Input>                          </FormGroup>                          <FormGroup>                              <Label for="user\_id">User ID</Label>                              <Input                                  id="user\_id"                                  value={this.state.updatePostData.user\_id}                                  onChange={(e) => {                                      let { updatePostData } = this.state                                      updatePostData.user\_id = e.target.value                                      this.setState({ updatePostData })                                  }}                              ></Input>                          </FormGroup>                      </ModalBody>                      <ModalFooter>                          <Button color="primary" onClick={this.updatePost.bind(this)}> Edit Post </Button>{' '}                          <Button color="secondary" onClick={this.toggleUpdatePostModal.bind(this)}> Cancel </Button>                      </ModalFooter>                  </Modal>                  <Table>                      <thead>                          <tr>                              <th>ID</th>                              <th>Title</th>                              <th>Content</th>                              <th>Actions</th>                          </tr>                      </thead>                      <tbody>                          {posts}                      </tbody>                  </Table>              </div>          );      }  }  if (document.getElementById('example')) {      ReactDOM.render(<Example />, document.getElementById('example'));  } |

**2. Using React Component within React Component.**

We can modularize the larger component into smaller components. Let’s modularize the “addPost” into another component as shown in Figure 13.

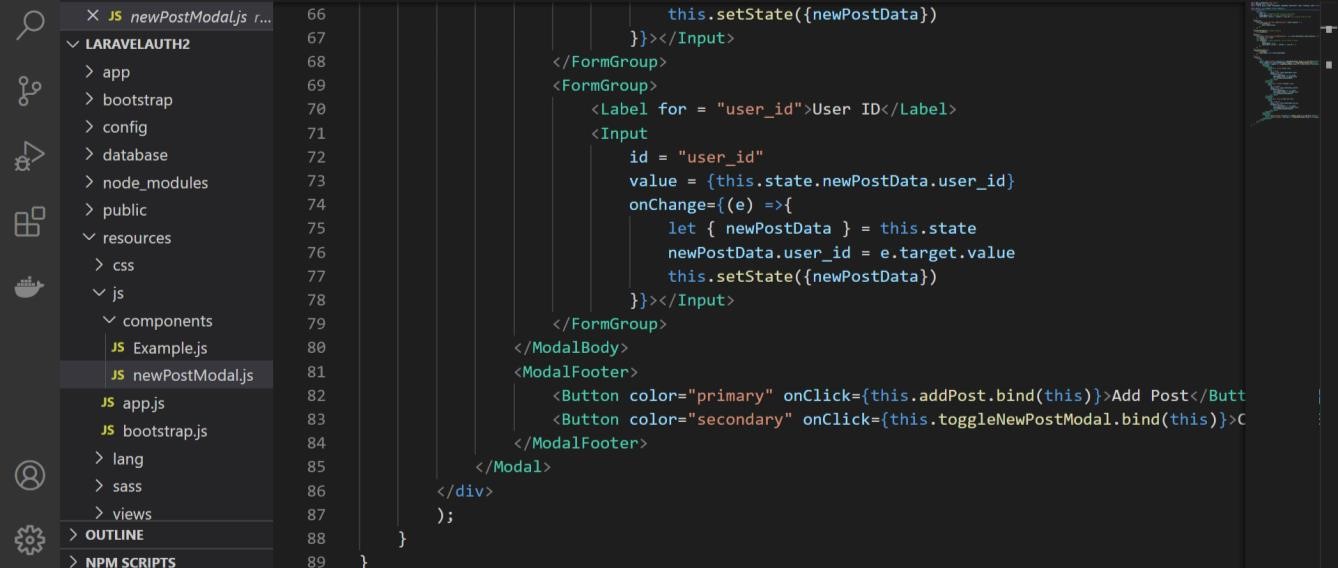
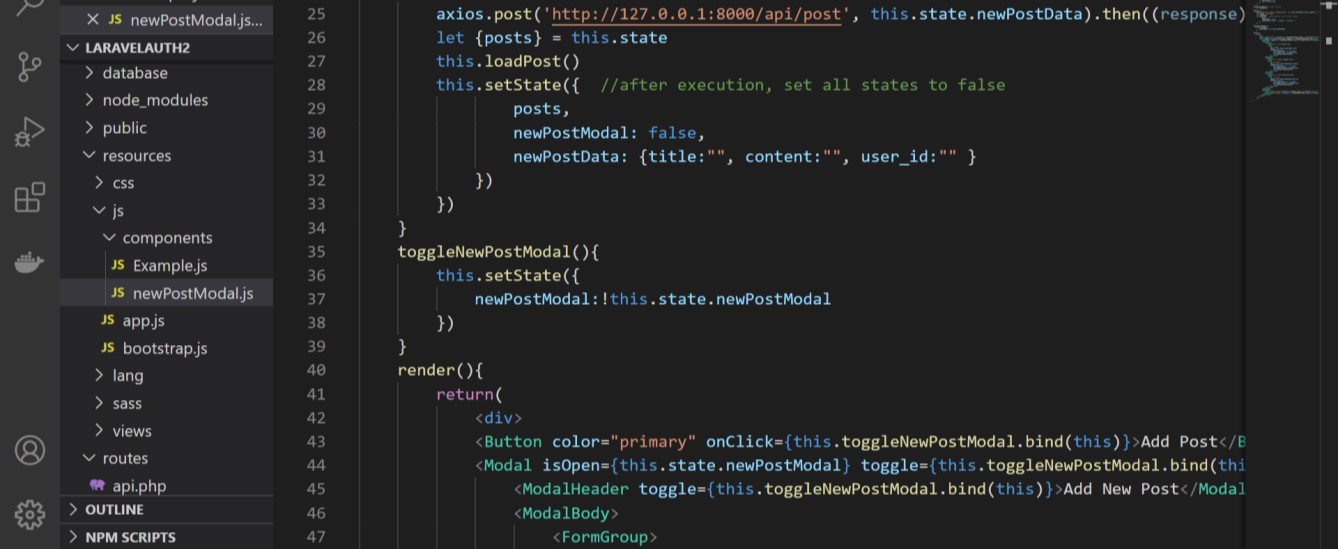
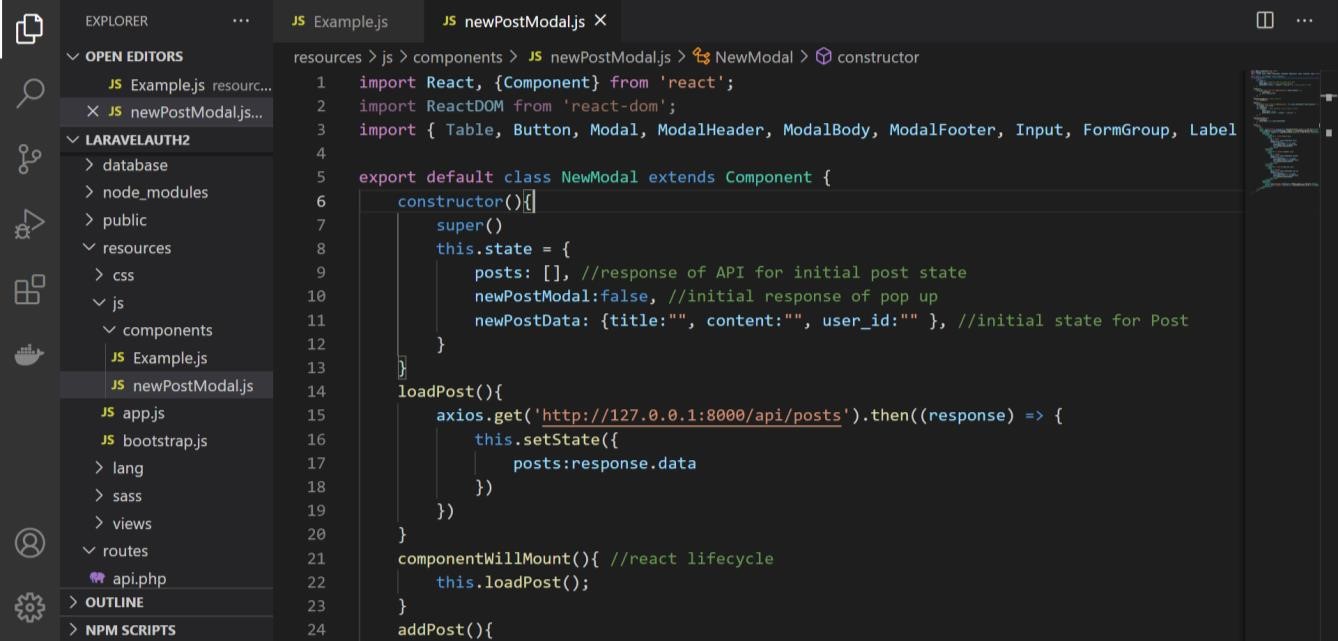


Figure 13: “addPost” React Component.

If there are troubles loading popper js or Sourc

eMaps (no modal after clicking “Add Post” button),

follow the steps below:

Within webpack.mix.js

Within resources

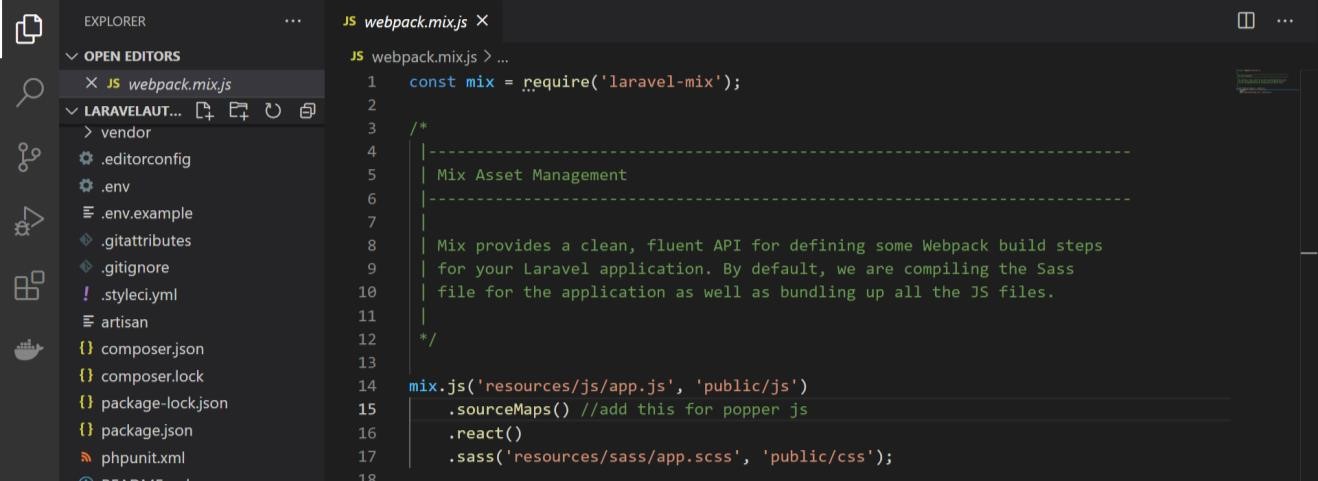
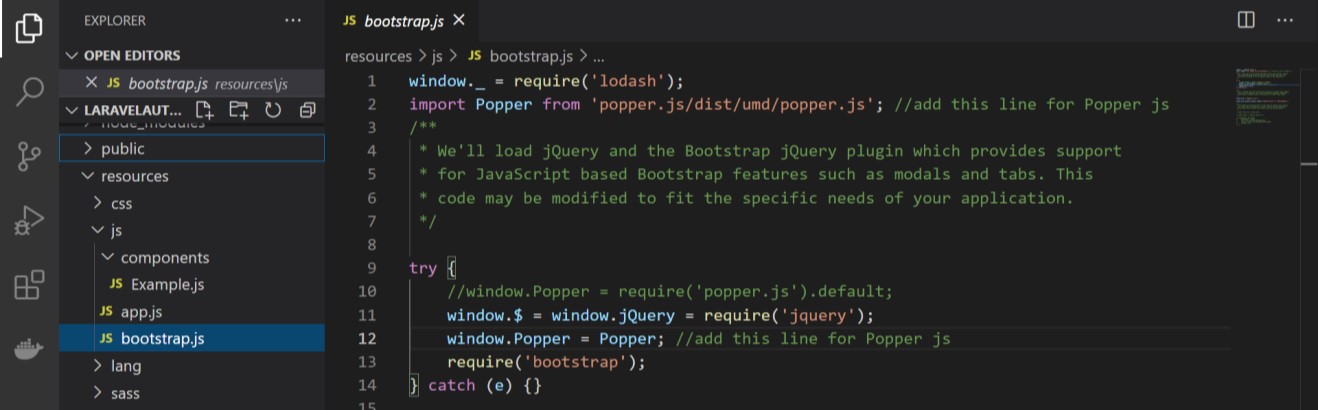
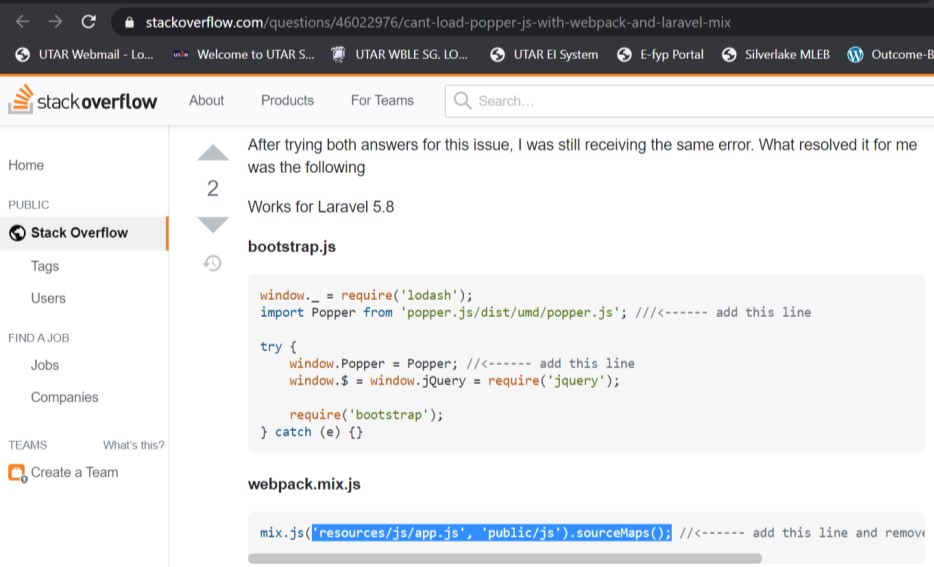
\

js

\

bootstrap.js

This works for enabling the popper js and SourceMaps.



UECS3294 ADVANCED WEB APPLICATION DEVELOPMENT

|  |
| --- |
| Example7.js  import React, { Component } from 'react';  import ReactDOM from 'react-dom';  import { Table, Button, Modal, ModalHeader, ModalBody, ModalFooter, FormGroup, Label, Input } from 'reactstrap';  import axios from 'axios';  export default class Example extends Component {      constructor() {          super()          this.state = {              posts: [], //response of API into post state              newPostModal: false,              updatePostModal: false,              newPostData: { title: "", content: "", user\_id: "" },              updatePostData: { id:"", title: "", content:"", user\_id:"" },              loginModal: false,              loginData: { email: "", password: ""},              loginResponseData:[]          }      }        //sessionStorage.setItem('token', token); var token = sessionStorage.getItem('token');      login() {          axios.post('http://127.0.0.1:8000/api/auth/login',          this.state.loginData).then((response) => {              let { loginResponseData } = this.state              this.setState({                  loginResponseData: response.data,                  loginModal: false,                  loginData: { email: "", password: ""}              })              this.loadPost()              console.warn(response.data);          })      }      toggleloginModal() {          this.setState({ loginModal: !this.state.loginModal })      }          loadPost() {          if (!this.state.loginResponseData.access\_token=="")          {              axios.post('http://127.0.0.1:8000/api/auth/index', {}, {               headers: {'Authorization':                  `Bearer ${this.state.loginResponseData.access\_token}`}              }).then((response) => {              this.setState({                      posts: response.data                  })              })          }      }        addPost() {          axios.post('http://127.0.0.1:8000/api/auth/store',          this.state.newPostData, {              headers: {                  'Authorization': `Bearer ${this.state.loginResponseData.access\_token}`              }          }).then((response) => {              let { posts } = this.state              this.loadPost()              this.setState({                  posts,                  newPostModal: false,                  newPostData: { title: "", content: "", user\_id: "" },                })          })      }          componentWillMount() {          this.loadPost();      }      toggleNewPostModal() {          this.setState({ newPostModal: !this.state.newPostModal })        }          callUpdatePost(id, title, content, user\_id) {          this.setState({              updatePostData: { id, title, content, user\_id },              updatePostModal: !this.state.updatePostModal          })      }        updatePost() {          let { id, title, content, user\_id } = this.state.updatePostData          axios.put('http://127.0.0.1:8000/api/auth/update/' + this.state.updatePostData.id, {              title, content, user\_id},              {       headers: {                      'Authorization': `Bearer ${this.state.loginResponseData.access\_token}`                  }              }).then((response) => {              this.loadPost()              this.setState({ //after execution, set all states to false                  updatePostModal: false,                  updatePostData: { id: "", title: "", content: "", user\_id: "" }              })          })      }      deletePost(id) {            if (confirm("Do you want delete this Post?"))          {          axios.delete('http://127.0.0.1:8000/api/post/' + id, { },          {              headers: {                  'Authorization': `Bearer ${this.state.loginResponseData.access\_token}`              }          }).then((response) => {              this.loadPost()          })      }      }      toggleUpdatePostModal() {          this.setState({              updatePostModal: !this.state.updatePostModal          })      }      render() {          let posts = this.state.posts.map((post) => {              return (                  <tr key={post.id}>                      <td>{post.id}</td>                      <td>{post.title}</td>                      <td>{post.content}</td>                      <td>{post.user\_id}</td>                      <td>                          <Button color="success" size="sm" className="mr-2"                              onClick={this.callUpdatePost.bind(this,post.id, post.title, post.content, post.user\_id)}>                              Edit </Button>                          <Button color="danger" size="sm" className="mr-2"  onClick={this.deletePost.bind(this,post.id)}> Delete </Button>                      </td>                  </tr>              )          })          return (              <div className="container">                  <Button color="primary" onClick={this.toggleloginModal.bind(this)}>Login</Button>                  {"      "}                  <Button color="primary" onClick={this.toggleNewPostModal.bind(this)}>Add Post</Button>                  <Modal isOpen={this.state.newPostModal} toggle={this.toggleNewPostModal.bind(this)}>                      <ModalHeader toggle={this.toggleNewPostModal.bind(this)}> Add New Post                      </ModalHeader>                      <ModalBody>                          <FormGroup>                              <Label for="title">Title</Label>                              <Input                                  id="title"                                  value={this.state.newPostData.title}                                  onChange={(e) => {                                      let { newPostData } = this.state                                      newPostData.title = e.target.value                                      this.setState({ newPostData })                                  }}                              ></Input>                          </FormGroup>                          <FormGroup>                              <Label for="content">Content</Label>                              <Input                                  id="content"                                  value={this.state.newPostData.content}                                  onChange={(e) => {                                      let { newPostData } = this.state                                      newPostData.content = e.target.value                                      this.setState({ newPostData })                                  }}                              ></Input>                          </FormGroup>                          <FormGroup>                              <Label for="user\_id">User ID</Label>                              <Input                                  id="user\_id"                                  value={this.state.newPostData.user\_id}                                  onChange={(e) => {                                      let { newPostData } = this.state                                      newPostData.user\_id = e.target.value                                      this.setState({ newPostData })                                  }}                              ></Input>                          </FormGroup>                      </ModalBody>                      <ModalFooter>                          <Button color="primary" onClick={this.addPost.bind(this)}> Add Post </Button>{' '}                          <Button color="secondary" onClick={this.toggleNewPostModal.bind(this)}> Cancel </Button>                      </ModalFooter>                  </Modal>                    <Modal isOpen={this.state.updatePostModal} toggle={this.toggleUpdatePostModal.bind(this)}>                      <ModalHeader toggle={this.toggleUpdatePostModal.bind(this)}>Upadte Post                      </ModalHeader>                      <ModalBody>                          <FormGroup>                              <Label for="title">Title</Label>                              <Input                                  id="title"                                  value={this.state.updatePostData.title}                                  onChange={(e) => {                                      let { updatePostData } = this.state                                      updatePostData.title = e.target.value                                      this.setState({ updatePostData })                                  }}                              ></Input>                          </FormGroup>                          <FormGroup>                              <Label for="content">Content</Label>                              <Input                                  id="content"                                  value={this.state.updatePostData.content}                                  onChange={(e) => {                                      let { updatePostData } = this.state                                      updatePostData.content = e.target.value                                      this.setState({ updatePostData })                                  }}                              ></Input>                          </FormGroup>                          <FormGroup>                              <Label for="user\_id">User ID</Label>                              <Input                                  id="user\_id"                                  value={this.state.updatePostData.user\_id}                                  onChange={(e) => {                                      let { updatePostData } = this.state                                      updatePostData.user\_id = e.target.value                                      this.setState({ updatePostData })                                  }}                              ></Input>                          </FormGroup>                      </ModalBody>                      <ModalFooter>                          <Button color="primary" onClick={this.updatePost.bind(this)}> Edit Post </Button>{' '}                          <Button color="secondary" onClick={this.toggleUpdatePostModal.bind(this)}> Cancel </Button>                      </ModalFooter>                  </Modal>                      <Modal isOpen={this.state.loginModal} toggle={this.toggleloginModal.bind(this)}>                      <ModalHeader toggle={this.toggleloginModal.bind(this)}> Login                      </ModalHeader>                      <ModalBody>                          <FormGroup>                              <Label for="email">Email</Label>                              <Input                                  id="email"                                  value={this.state.loginData.email}                                  onChange={(e) => {                                      let { loginData } = this.state                                      loginData.email = e.target.value                                      this.setState({ loginData })                                  }}                              ></Input>                          </FormGroup>                          <FormGroup>                              <Label for="password">Password</Label>                              <Input                                  id="password"                                  type="password"                                  value={this.state.loginData.password}                                  onChange={(e) => {                                      let { loginData } = this.state                                      loginData.password = e.target.value                                      this.setState({ loginData })                                  }}                              ></Input>                          </FormGroup>                        </ModalBody>                      <ModalFooter>                          <Button color="primary" onClick={this.login.bind(this)}> Login </Button>{'    '}                          <Button color="secondary" onClick={this.toggleloginModal.bind(this)}> Cancel </Button>                      </ModalFooter>                  </Modal>                  <Table>                      <thead>                          <tr>                              <th>Id</th>                              <th>Title</th>                              <th>Content</th>                              <th>User id</th>                              <th>Actions</th>                          </tr>                      </thead>                      <tbody>                          {posts}                      </tbody>                    </Table>              </div>          );      }  }    if (document.getElementById('example')) {      ReactDOM.render(<Example />, document.getElementById('example'));  } |