Setup Frontend

|  |  |
| --- | --- |
| Open command prompt and navigate to your Projects folder created in the previous module |  |
| Create a new folder named frontend | Md frontend |
| Change directory to frontend | Cd frontend |
| Create fabric-ui app | npm init uifabric |
| Open vscode | code . |
| Start web page | npm start |
|  |  |
|  |  |
|  |  |
|  |  |

Bikes component

|  |  |
| --- | --- |
| Create Components folder in vs code (under src) |  |
| Create new file Bikes.tsx |  |
| Add stub code | import React, { FunctionComponent } from "react";  export default class Bikes extends React.Component  {      componentDidMount()      {        }      render() {              return (              <div>                 Hello bikes!              </div>          );      }  } |
| Change Index.tsx to point to new bike component | ReactDOM.render(<Bikes />,  document.getElementById('app')); |
| Hello Bikes renders |  |

Classes

|  |  |
| --- | --- |
| Create folder Models |  |
| Create Bike.tsx class | class Bike {      id: string;      model: string;      make: string;      price: number;      quantity: number;        constructor()      {          this.id = "";          this.model = "";          this.make = "";          this.price = 0;          this.quantity = 0;      }    }    export default Bike |

Modify app to show data

|  |  |
| --- | --- |
| Create interface | interface IBikes {      m\_aBike: Bike[];    } |
| Define stack item style | const itemStyles: React.CSSProperties = {      alignItems: 'start',      display: 'flex',      justifyContent: 'start',      width: 150,      padding: 10    }; |
| Define header label style | const lblStyle: React.CSSProperties = {     fontSize: 20    }; |
| Modify component definition | export default class Bikes  extends React.Component<{  }, IBikes> |
| Prepare constructor | constructor(props: Readonly<{}>) {          super(props);            this.state =          {            m\_aBike: new Array()          };        } |
| Modify componentDidMount place your URL from the Bikes function | componentDidMount()      {          fetch(FUNCTION URL)              .then(res => res.json())              .then(res => {                   this.setState({ m\_aBike: res });          });      } |
| Create a function to emit a row | RenderBike(b: Bike)      {          var formatter = new Intl.NumberFormat('en-US', {              style: 'currency',              currency: 'USD',            });          return <Stack horizontal>              <Label style={itemStyles}>{b.make}</Label>              <Label style={itemStyles}>{b.model}</Label>              <Label style={itemStyles}>{b.make}</Label>              <Label style={itemStyles}>  {formatter.format(b.price)}</Label>              <Label style={itemStyles}>  {b.quantity}</Label>          </Stack>      } |
| Complete the render function | render() {              return (                  <Stack padding={10}>                      <Label style={lblStyle}>  Current Bikes</Label>                      <Stack>                          {this.state.m\_aBike.map  (v => this.RenderBike(v))}                      </Stack>                  </Stack>          );      } |

Add edit Quantity

|  |  |
| --- | --- |
| Add variables for editing (selected bike and new quantity and edit open) | interface IBikes {      m\_aBike: Bike[];      m\_bEditOpen: boolean;      m\_SelectedBike: Bike;      m\_nNewQuantity: number;    } |
| Add style for panel | const pnlStyle: React.CSSProperties = {      margin:5,      width: 200     }; |
| Set initial state (in constructor) | this.state =          {            m\_aBike: new Array(),            m\_bEditOpen: false,            m\_SelectedBike: new Bike(),            m\_nNewQuantity: 0          }; |
| Set bindings (in constructor) | this.QuantityChange = this.QuantityChange.bind(this); |
| Create show edit function | ShowEdit(b: Bike)      {          this.setState  ({m\_SelectedBike: b,  m\_nNewQuantity: b.quantity,  m\_bEditOpen: true});      } |
| Create close edit function | CloseEdit()      {          this.setState  ({m\_bEditOpen: false});      } |
| Create the save function | Save()      {          let v: Bike =  this.state.m\_SelectedBike;          v.quantity =  this.state.m\_nNewQuantity;          fetch(BIKE URL, {              method: 'post',              body: JSON.stringify(v)          }).then(res => {              window.location.href = '/';          }).catch(error => alert('Error! '   + error.message));          this.setState({m\_bEditOpen: false});      } |
| Create quantity change function | QuantityChange = (event: any)=> {          this.setState  ({ m\_nNewQuantity: event.target.value });        } |
| Add edit button | RenderBike(b: Bike)      {          var formatter =  new Intl.NumberFormat('en-US', {              style: 'currency',              currency: 'USD',            });          return <Stack horizontal>              <Label style={itemStyles}>{b.make}</Label>              <Label style={itemStyles}>{b.model}</Label>              <Label style={itemStyles}>{b.make}</Label>              <Label style={itemStyles}>  {formatter.format(b.price)}</Label>              <Label style={itemStyles}>  {b.quantity}</Label>              <PrimaryButton text="Edit"   onClick={() => this.ShowEdit(b)}  allowDisabledFocus />          </Stack>      } |
| Added edit panel component | <Panel    headerText={this.state.m\_SelectedBike.model}    isOpen={this.state.m\_bEditOpen}    closeButtonAriaLabel="Close">    <Stack>    <TextField style={pnlStyle} label="Quantity"  onChange={this.QuantityChange}  value={this.state.m\_nNewQuantity.toString()  } />    <PrimaryButton style={pnlStyle} text="Save"  onClick={() => this.Save()} allowDisabledFocus />    <PrimaryButton style={pnlStyle}  text="Close" onClick={() => this.CloseEdit()}  allowDisabledFocus />                          </Stack>                      </Panel> |

Add “Add”

|  |  |
| --- | --- |
| Add variable for panel | interface IBikes {      m\_aBike: Bike[];      m\_bEditOpen: boolean;      m\_SelectedBike: Bike;      m\_nNewQuantity: number;      m\_bAddOpen: boolean;    } |
| Set in constructor | this.state =          {            m\_aBike: new Array(),            m\_bEditOpen: false,            m\_SelectedBike: new Bike(),            m\_nNewQuantity: 0,            m\_bAddOpen: false          }; |
| Set bindings (in constructor) | this.AddMakeChange = this.AddMakeChange.bind(this);  this.AddModelChange = this.AddModelChange.bind(this);  this.AddPriceChange = this.AddPriceChange.bind(this);  this.AddQuantityChange = this.AddQuantityChange.bind(this); |
| Add add handlers | AddMakeChange =  (event: any)   => {          let v: Bike =  this.state.m\_SelectedBike;          v.make = event.target.value;          this.setState({ m\_SelectedBike: v });      }        AddModelChange =   (event: any)  => {          let v: Bike =  this.state.m\_SelectedBike;          v.model = event.target.value;          this.setState({ m\_SelectedBike: v });      }      AddPriceChange =  (event: any) => {          let v: Bike =  this.state.m\_SelectedBike;          v.price = event.target.value;          this.setState({ m\_SelectedBike: v });      }      AddQuantityChange =   (event: any)=> {          let v: Bike =  this.state.m\_SelectedBike;          v.quantity = event.target.value;          this.setState({ m\_SelectedBike: v });      } |
| Add close Add panel | CloseAdd()      {          this.setState({m\_bAddOpen: false});      } |
| Add show Add panel | ShowAdd()      {          this.setState({m\_bAddOpen: true});      } |
| Add save function | Add()      {          fetch(BIKE URL, {              method: 'post',              body: JSON.stringify  (this.state.m\_SelectedBike)          }).then(res => {              window.location.href = '/';          }).catch(error =>  alert('Error! ' + error.message));          this.setState({m\_bAddOpen: false});      } |
| Add Edit Panel | <Panel     headerText="New Bike"  isOpen={this.state.m\_bAddOpen}    closeButtonAriaLabel="Close">  <Stack>    <TextField style={pnlStyle} label="Make"  onChange={this.AddMakeChange}/>    <TextField style={pnlStyle} label="Model"  onChange={this.AddModelChange}/>    <TextField style={pnlStyle} label="Price"  onChange={this.AddPriceChange}/>    <TextField style={pnlStyle} label="Quantity"  onChange={this.AddQuantityChange}/>    <PrimaryButton style={pnlStyle} text="Add"  onClick={() => this.Add()} allowDisabledFocus />    <PrimaryButton style={pnlStyle} text="Close"  onClick={() => this.CloseAdd()} allowDisabledFocus />                          </Stack>                      </Panel> |
| Add “Add” button | <PrimaryButton style={pnlStyle}  text="Add" onClick={() => this.ShowAdd()}    allowDisabledFocus /> |