// src/components/ImageGrid.js

Import React, { Component } from ‘react’;

Import { Container, Row, Col, Modal, ModalHeader, ModalBody, Button } from ‘reactstrap’;

Import one from ‘../assets/one.jpg’;

Import two from ‘../assets/two.jpg’;

Import three from ‘../assets/three.jpg’;

Import four from ‘../assets/four.jpg’;

Import five from ‘../assets/five.jpg’;

Import six from ‘../assets/six.jpg’;

Import seven from ‘../assets/seven.jpg’;

Import eight from ‘../assets/eight.jpg’;

Import nine from ‘../assets/nine.avif’;

Import ten from ‘../assets/ten.jpg’;

Import eleven from ‘../assets/eleven.jpg’;

Import tweleve from ‘../assets/tweleve.webp’;

Import ImageCard from ‘./ImageCard’;

Const object = {

9: {

0: {

“startTime”: ‘9:00’,

“endTime”: “9:05”,

File: one

},

1: {

“startTime”: ‘9:05’,

“endTime”: “9:10”,

File: two

},

2: {

“startTime”: ‘9:10’,

“endTime”: “9:15”,

File: three

},

3: {

“startTime”: ‘9:15’,

“endTime”: “9:20”,

File: four

},

4: {

“startTime”: ‘9:20’,

“endTime”: “9:25”,

File: five

},

5: {

“startTime”: ‘9:25’,

“endTime”: “9:30”,

File: six

},

6: {

“startTime”: ‘9:30’,

“endTime”: “9:35”,

File: seven

},

7: {

“startTime”: ‘9:35’,

“endTime”: “9:40”,

File: eight

},

8: {

“startTime”: ‘9:40’,

“endTime”: “9:45”,

File: nine

},

9: {

“startTime”: ‘9:45’,

“endTime”: “9:50”,

File: ten

},

10: {

“startTime”: ‘9:50’,

“endTime”: “9:55”,

File: eleven

},

11: {

“startTime”: ‘9:55’,

“endTime”: “10:00”,

File: tweleve

},

},

10: {

0: {

“startTime”: ’10:00’,

“endTime”: “10:05”,

File: one

},

1: {

“startTime”: ’10:05’,

“endTime”: “10:10”,

File: two

},

2: {

“startTime”: ’10:10’,

“endTime”: “10:15”,

File: three

},

3: {

“startTime”: ’10:15’,

“endTime”: “10:20”,

File: four

},

4: {

“startTime”: ’10:20’,

“endTime”: “10:25”,

File: five

},

5: {

“startTime”: ’10:25’,

“endTime”: “10:30”,

File: six

},

6: {

“startTime”: ’10:30’,

“endTime”: “10:35”,

File: seven

},

7: {

“startTime”: ’10:35’,

“endTime”: “10:40”,

File: eight

},

8: {

“startTime”: ’10:40’,

“endTime”: “10:45”,

File: nine

},

9: {

“startTime”: ’10:45’,

“endTime”: “10:50”,

File: ten

},

10: {

“startTime”: ’10:50’,

“endTime”: “10:55”,

File: eleven

},

11: {

“startTime”: ’10:55’,

“endTime”: “11:00”,

File: tweleve

},

},

11: {

0: {

“startTime”: ’11:00’,

“endTime”: “11:05”,

File: one

},

1: {

“startTime”: ’11:05’,

“endTime”: “11:10”,

File: two

},

2: {

“startTime”: ’11:10’,

“endTime”: “11:15”,

File: three

},

3: {

“startTime”: ’11:15’,

“endTime”: “11:20”,

File: four

},

4: {

“startTime”: ’11:20’,

“endTime”: “11:25”,

File: five

},

5: {

“startTime”: ’11:25’,

“endTime”: “11:30”,

File: six

},

6: {

“startTime”: ’11:30’,

“endTime”: “11:35”,

File: seven

},

7: {

“startTime”: ’11:35’,

“endTime”: “11:40”,

File: eight

},

8: {

“startTime”: ’11:40’,

“endTime”: “11:45”,

File: nine

},

9: {

“startTime”: ’11:45’,

“endTime”: “11:50”,

File: ten

},

10: {

“startTime”: ’11:50’,

“endTime”: “11:55”,

File: eleven

},

11: {

“startTime”: ’11:55’,

“endTime”: “12:00”,

File: tweleve

},

},

12: {

0: {

“startTime”: ’12:00’,

“endTime”: “12:05”,

File: one

},

1: {

“startTime”: ’12:05’,

“endTime”: “12:10”,

File: two

},

2: {

“startTime”: ’12:10’,

“endTime”: “12:15”,

File: three

},

3: {

“startTime”: ’12:15’,

“endTime”: “12:20”,

File: four

},

4: {

“startTime”: ’12:20’,

“endTime”: “12:25”,

File: five

},

5: {

“startTime”: ’12:25’,

“endTime”: “12:30”,

File: six

},

6: {

“startTime”: ’12:30’,

“endTime”: “12:35”,

File: seven

},

7: {

“startTime”: ’12:35’,

“endTime”: “12:40”,

File: eight

},

8: {

“startTime”: ’12:40’,

“endTime”: “12:45”,

File: nine

},

9: {

“startTime”: ’12:45’,

“endTime”: “12:50”,

File: ten

},

10: {

“startTime”: ’12:50’,

“endTime”: “12:55”,

File: eleven

},

11: {

“startTime”: ’12:55’,

“endTime”: “13:00”,

File: tweleve

},

},

13: {

0: {

“startTime”: ’13:00’,

“endTime”: “13:05”,

File: one

},

1: {

“startTime”: ’13:05’,

“endTime”: “13:10”,

File: two

},

2: {

“startTime”: ’13:10’,

“endTime”: “13:15”,

File: three

},

3: {

“startTime”: ’13:15’,

“endTime”: “13:20”,

File: four

},

4: {

“startTime”: ’13:20’,

“endTime”: “13:25”,

File: five

},

5: {

“startTime”: ’13:25’,

“endTime”: “13:30”,

File: six

},

6: {

“startTime”: ’13:30’,

“endTime”: “13:35”,

File: seven

},

7: {

“startTime”: ’13:35’,

“endTime”: “13:40”,

File: eight

},

8: {

“startTime”: ’13:40’,

“endTime”: “13:45”,

File: nine

},

9: {

“startTime”: ’13:45’,

“endTime”: “13:50”,

File: ten

},

10: {

“startTime”: ’13:50’,

“endTime”: “13:55”,

File: eleven

},

11: {

“startTime”: ’13:55’,

“endTime”: “14:00”,

File: tweleve

},

}

};

Class ImageGrid extends Component {

State = {

modalOpen: false,

currentImage: null,

currentHour: null,

currentIndex: null,

absoluteIndex: 0,

totalImages: 0

};

componentDidMount() {

window.addEventListener(‘keydown’, this.handleKeyDown);

this.calculateTotalImages();

}

componentWillUnmount() {

window.removeEventListener(‘keydown’, this.handleKeyDown);

}

calculateTotalImages = () => {

const totalImages = Object.keys(object).reduce((sum, hour) => {

return sum + Object.keys(object[hour]).length;

}, 0);

This.setState({ totalImages });

};

openModal = (hour, index) => {

const absoluteIndex = this.calculateAbsoluteIndex(hour, index);

this.setState({

modalOpen: true,

currentImage: object[hour][index].file,

currentHour: hour,

currentIndex: index,

absoluteIndex

});

};

closeModal = () => {

this.setState({ modalOpen: false });

};

handleKeyDown = (event) => {

if (this.state.modalOpen) {

if (event.key === ‘ArrowRight’) {

this.showNextImage();

} else if (event.key === ‘ArrowLeft’) {

This.showPrevImage();

}

}

};

showNextImage = () => {

const { currentHour, currentIndex, absoluteIndex, totalImages } = this.state;

let nextIndex = currentIndex + 1;

let nextHour = currentHour;

let newAbsoluteIndex = absoluteIndex + 1;

if (newAbsoluteIndex >= totalImages) {

newAbsoluteIndex = 0; // Loop back to the first image

}

If (nextIndex >= Object.keys(object[currentHour]).length) {

nextIndex = 0;

nextHour = currentHour + 1;

}

If (object[nextHour] && object[nextHour][nextIndex]) {

This.setState({

currentImage: object[nextHour][nextIndex].file,

currentHour: nextHour,

currentIndex: nextIndex,

absoluteIndex: newAbsoluteIndex

});

}

};

showPrevImage = () => {

const { currentHour, currentIndex, absoluteIndex, totalImages } = this.state;

let prevIndex = currentIndex – 1;

let prevHour = currentHour;

let newAbsoluteIndex = absoluteIndex – 1;

if (newAbsoluteIndex < 0) {

newAbsoluteIndex = totalImages – 1; // Loop back to the last image

}

If (prevIndex < 0) {

prevHour = currentHour – 1;

prevIndex = Object.keys(object[prevHour]).length – 1;

}

If (object[prevHour] && object[prevHour][prevIndex]) {

This.setState({

currentImage: object[prevHour][prevIndex].file,

currentHour: prevHour,

currentIndex: prevIndex,

absoluteIndex: newAbsoluteIndex

});

}

};

calculateAbsoluteIndex = (hour, index) => {

let absoluteIndex = 0;

for (let h = 9; h < hour; h++) {

absoluteIndex += Object.keys(object[h]).length;

}

absoluteIndex += index;

return absoluteIndex;

};

Render() {

Const { absoluteIndex, totalImages, modalOpen, currentImage } = this.state;

Return (

<Container>

{Object.keys(object).map(hour => (

<div key={hour} className=”hour-section”>

<h2>{`${hour}:00`}</h2>

<Row>

{Object.keys(object[hour]).map(index => (

<Col sm=”2” key={index}>

<ImageCard

Hour={hour}

Index={index}

Image={object[hour][index].file}

startTime={object[hour][index].startTime}

endTime={object[hour][index].endTime}

openModal={this.openModal}

/>

</Col>

))}

</Row>

</div>

))}

{modalOpen && (

<Modal isOpen={modalOpen} toggle={this.closeModal}>

<ModalHeader toggle={this.closeModal}>Image</ModalHeader>

<ModalBody>

<div className=”modal-image-container”>

<img src={currentImage} alt=”” className=”modal-image”/>

<div className=”modal-navigation”>

<Button color=”primary” onClick={this.showPrevImage}>Previous</Button>

<Button color=”primary” onClick={this.showNextImage}>Next</Button>

</div>

<div className=”image-index”>

{`${absoluteIndex + 1} / ${totalImages}`}

</div>

</div>

</ModalBody>

</Modal>

)}

</Container>

);

}

}

Export default ImageGrid;