SUNSTONE

Calendar Scheduler App

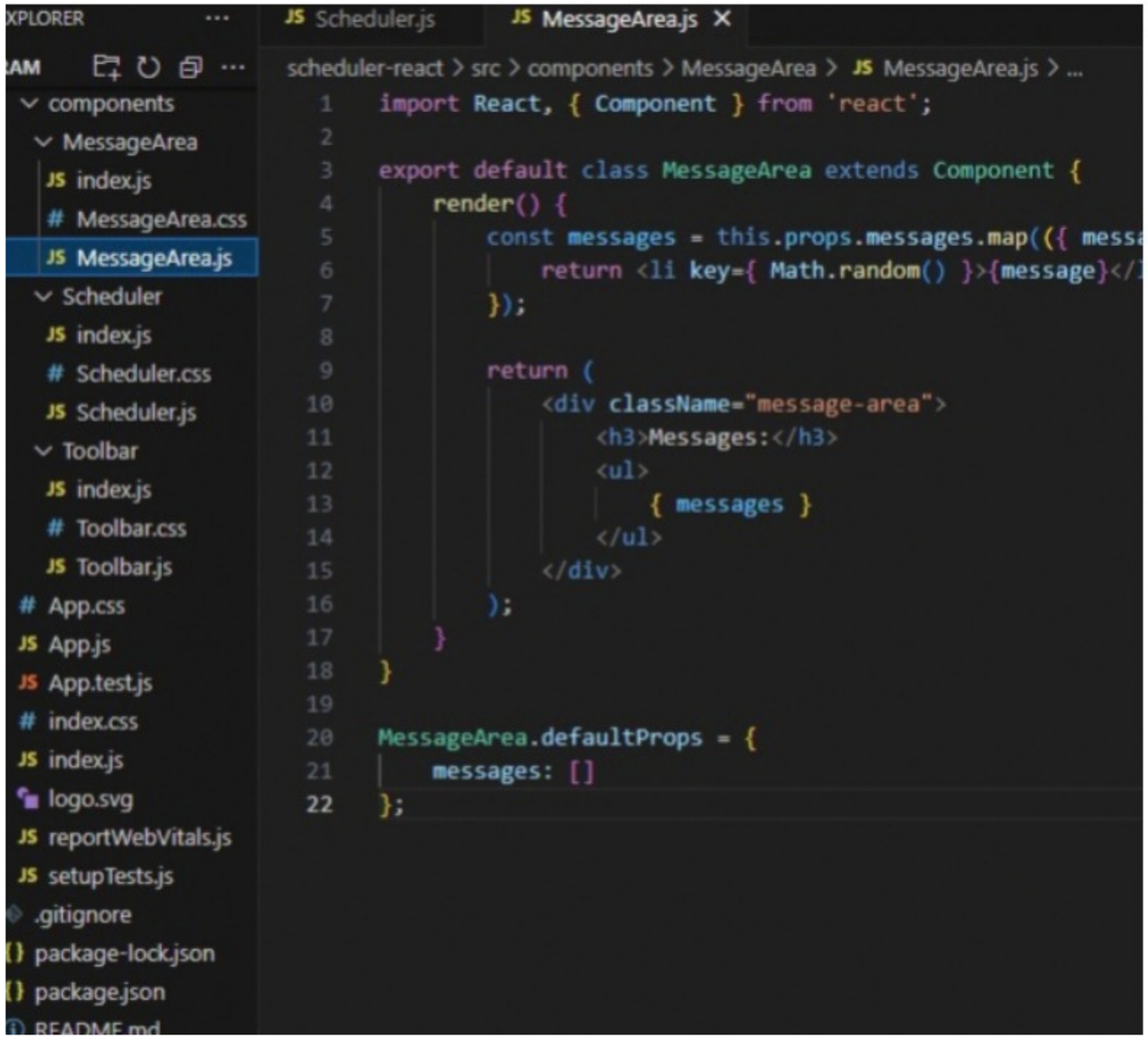
Task - 3

Calender Scheduler APP

Schedumate is a user-friendly calendar scheduler app that helps you effortlessly manage your schedule and appointments. Key features include easy scheduling, cross-device syncing, smart reminders, customization, collaboration, to-do lists, event suggestions, weather integration, and robust data security.

LMS USERNAME	NAME	BATCH
au910020104011	DEEPTHIPRIYA R	CC2
au910020104012	DHARSHIKA R S	CC2
au910020104023	LAKSHMANA PRIYA A	CC2
au910020104310	SANTHOSH SIVAN S	CC2

Front End :


components
MessageArea
index.js
MessageArea.css
MessageArea.js
Scheduler
index.js
Scheduler.css
Scheduler.js
Toolbar
index.js
Toolbar.css
Toolbar.js
App.css
App.js
App.test.js
index.css
index.js
logo.svg
reportWebVitals.js
setupTests.js
.gitignore
package-lock.json
package.json
Scheduler.js
MessageArea.js
scheduler-react > src > components > MessageArea > MessageArea.js > ...
import React, { Component } from 'react';
export default class MessageArea extends Component {
render() {
const messages = this.props.messages.map(({ mess
return <li key={ Math.random() }>{message}</
});
return (

<h3>Messages:</h3>

{ messages }

);
}
}
MessageArea.defaultProps = {
messages: []
};


Scheduler.js
scheduler-react > src > components > Scheduler > Scheduler.js > Scheduler > initSc
import React, { Component } from 'react';
import 'dhtmlx-scheduler';
import 'dhtmlx-scheduler/codebase/dhtmlxscheduler_material.css';
const scheduler = window.scheduler;
export default class Scheduler extends Component {
initSchedulerEvents() {
if (scheduler._$initialized) {
return;
}
const onDataUpdated = this.props.onDataUpdated;
scheduler.attachEvent('onEventAdded', (id, ev) => {
if (onDataUpdated) {
onDataUpdated('create', ev, id);
}
});
scheduler.attachEvent('onEventChanged', (id, ev) => {
if (onDataUpdated) {
onDataUpdated('update', ev, id);
}
});

Highlighting the changes implemented :

Messages:

```
> event create: 1700053808787 (DIWALI)
> event delete: 1700053808785
> event update: 1700053808785
> event create: 1700053808785 (vinayagar)
```

Evaluation Metrics:

- **User Engagement :** Monitor time spent and interactions per session.
- **Performance :** Optimize page load times and rendering speed.
- **Responsiveness Across Devices :** Ensure consistent display on various screen sizes.
- **Feature Adoption :** Analyze usage frequency for key features.
- **Error Rate and User Feedback :** Address errors promptly, gather user feedback.
- **Accessibility :** Test and improve accessibility features.
- **Conversion Rates :** Analyze user completion of desired actions.
- **Cross-Browser Compatibility :** Ensure uniform rendering across different browsers.
- **User Retention :** Assess returning user numbers over time.
- **Loading and Navigation Metrics :** Optimize loading times and assess navigation ease.

Step-Wise Description :

- **Structure Components :** Organize components with separate files for code and styling.
- **Logic Implementation :** Develop functions, manage state, and handle interactions.
- **Styling :** Apply styles using CSS, following design principles.
- **Testing :** Write unit tests to ensure component functionality.
- **Documentation :** Add comments for clarity and create READMEs for reuse.
- **Optimization :** Integrate components, optimize for performance.
- **Accessibility and Usability :** Implement features for accessibility, conduct usability testing.
- **Cross-Browser Testing :** Test components on various browsers for consistency.
- **Error Handling :** Implement robust error handling mechanisms.
- **User Engagement Monitoring :** Track user engagement metrics for insights into user satisfaction.

Task Summary :

The front-end development of the Calendar Scheduler app focuses on organized component structures, logical functionality, and user-friendly styling. Rigorous testing and documentation ensure code reliability and clarity. Integration optimizes responsiveness and loading times, while accessibility and usability testing guarantee a positive user experience. Cross-browser compatibility is verified, and robust error handling mechanisms are in place. Ongoing improvements are driven by user feedback, with continuous monitoring of user engagement metrics for insights and enhancements.

Submission Github

GitHub Link

Thank you!