

## Calendar React Web App Documentation:

To use the app, user can add events by clicking on the 'Add Event' button. Events can be added to the calendar based on the time and date that user chooses. Each event has a start date and time and an end date and time. User can change the theme (mode) of the page by clicking on the 'Dark Mode' button. By clicking on the arrows on the right hand side of the page, user can navigate to the next and previous month. Also, if the day that user has selected is not today, by clicking on the 'today' button, it will navigate to the day today.

This app has both server and client sides. On the client side, we have Components and App.js

( and other css and js files). Components folder includes three components: addEventModal.jsx, Calendar.jsx and EditEvent.jsx. these components give app necessary functionality and structure. Most functionality of the app is coded (implemented) in the Calendar.jsx . State has been used in these components . As mentioned above, app has the Addin Event functionality, changing the mode to the dark mode, navigating to the next, previous month and today. Events are added based on the users' choice on date and time. In Calendar.jsx, API, POST and PUT has been used as well as axios to make requests. For catching possible errors try and catch has been used in functions handleEventUpdate and handleEventAdd, to catch the error and display them in the console with an error message. Also, to store the dark mode local storage has been used.

Moreover, this app tries to connect to MondoDB to store the database, so we can have dynamic database. MongoDB URI is added to the .env file this is done in the server side. In fact in the server side, we have server.js which takes necessary requires (express, mongoose and body-parser), tries to connect to MongoDB and if the app connects to MongoDB successfully, it will show a message in the console to let the user know also we listen to the port 5000 in the server.js. Then we have controllers folder in the server side, which contains CalendarController.js , which we have router.post and router.get here . Then we have Models folder which contains Event.js , which tries to create EventSchema ( an object which is in fact the event)