Calendar+

Authors: Chad Kinnard, Bo Yang Li, Tyler Marrapese, Warren Overstreet

Changes and Deviations

Node Modules

As our understanding of Node.js has considerably improved since the project requirements document submission and also our understanding of what we need in code for the application to work has improved, We have included a few modules that we did not mention before as we did not know they were necessary. The Express module for node.js is a server-side web application framework which is required for us to have the server communicate with a client's webpage. The Morgan module for node.js is middleware to log HTTP requests and errors. The Bodyparser module for node.js allows the Express module to read HTTP POST requests. The MySQL module for node.js is quite straightforward and provides functions that allow us to interact with the server's SQL database. The Express-session module for node.js is a server-side framework that allows the server to track a client's session and also allows the server to distinguish clients. The Nodemon module for node.js is simply a tool to assist in the development of our application and is not required for the website to function, I figured it was still necessary to list the tools we are using.

GUI

The login and registration pages have been combined into one page through the use of JavaScript DOM elements. This change was not necessary, but it felt like a good change, as since they use the same styling and structure, it made more sense to have them on the same page rather than two separate pages.

Complete or Partially Complete

GUI

All forms and buttons are implemented and have requirements on the login and registration page. If the fields do not meet a field's requirements, they are notified through a popup under the input field. The month calendar components, which are a set of functions used in any view that contains a month calendar, are fully implemented. All buttons for the top nav bar have been created. There is a dropdown menu to select a different view; if the user tries to select the current view, it does nothing as the link for the current view is disabled. The login button is implemented, although it currently is not functional (it will be very easy to make it log the user out once the function has been implemented).

The "Create" button for the main view of the sidebar has been implemented, and when clicked, brings the user to the Event Creation view of the sidebar, which has also been implemented. Under the "Create" button, there is a mini calendar which is locked to the month view. It currently does function properly, as it reuses components from the actual month view calendar. However, it currently cannot be changed which month is being displayed (though, this is an easy fix). Another issue with it is that clicking on a date does not currently bring the main calendar component to that date. The Event Creation view contains multiple input fields and buttons, which don't currently have any function but will be implemented as the project progresses.

Styling has been applied as the pages have been created and developed, so for the most part, styling is in somewhat of a working and "final" state, with the exception of a few areas which are not finalized.

Account/Login System

Users can now enter their information to the database. This pulls the information from the input form on the registration page, then pushes it to the database. Once this is done, the user is redirected to the month view (this is the default view). Likewise, the user can also sign in to the account as long as the account has previously been registered in the database. If the username or password is incorrect, then as of right now, an alert will popup and deter the user back to the initial login screen. The rest of the restrictions are done through HTML elements such as username and password length.

We have decided on the “Express” framework which allows the server to easily accept <form> elements’ inputs. This is done through several POST routes. These routes include /user and /login which represent the registrationForm and loginForm respectively (found in the login.js file).

Event Creation

Like the Account and Login system, events are stored in the database. While not everything is figured out, things such as the incrementing event id, current user’s username, start and end time, and description are stored in the database. This is done the same as the Account/Login system. It uses “Express” to use the <form> elements in order to retrieve the input fields for the database. This form can be found in sidebar.js under the function initializeCreate().

Still Needs to be Implemented

GUI

On the login and registration page, it is currently planned to have a red outline around incorrect input fields. This will act as another way to indicate the input field contains incorrect information. Such as, when attempting to press the “Log In” button and a username or password is incorrect and/or not in the database.

Currently, the week view still needs to be implemented. The component for the day view should be able to be used as a backbone to render the week view, so this shouldn’t end up being too much of an issue or hassle to implement. The schedule view also needs to be implemented, which should be easy enough once there are events that can be pulled from the database.

Another item that’s still needed to be implemented, and will probably end up being the largest undertaking, is displaying events on the calendars. Currently, it is planned to do this through the use of a presentation layer to allow events to cross over from one date or time to another, but it’s unknown exactly how this will be implemented.

The sidebar isn't currently collapsible, as suggested in the design document. This is still planned to be added, but it is not very high up on the priority list. The "Event Information" sidebar view also still needs to be implemented.

Events/Reminders

As the backend is still being implemented, event reminders are yet to be implemented. For events, there is still the issue of adding the preferred color to the event. This is the color that will highlight the event on the calendar views for organizational use. Also, the event reminder time must be implemented.

Issues

Displaying Events Correctly

The frontend is required to communicate to the backend which events to display according to the user-selected timeframe. However, we currently do not have an ExpressJS route for event display on the calendar. Adding to the problem is that our application, as currently standing, relies on the frontend exclusively in order to change calendar views. This makes it more difficult for the backend to send the correct events for display. One potential solution would be to ensure that calendar views are served via the backend as ExpressJS GET routes.

Reminders

Since we have not fully integrated event creation, we have not quite gotten to the reminder system yet. The way we are currently looking to implement this is through the 'web-push' library. This provides both public and private VAPID keys which allows for push notifications. This could be problematic when trying to tie into the database, so this might not be the route we go with for the final product.

Time

Time can also be an issue. We have not necessarily stayed true to our original schedule. We need to manage time accordingly. Balance between this project and other classes could possibly be an issue as finals creep up. We need to save time for errors.

Hosting Server

We have currently been testing everything on our local machines, and have not pushed anything to the actual linode server yet. As of right now, we just have a test file that shows the server works. While this will come at a later date, and should be relatively straight-forward, we still need to account for the hiccups that could arise during this process.

Updated Diagrams

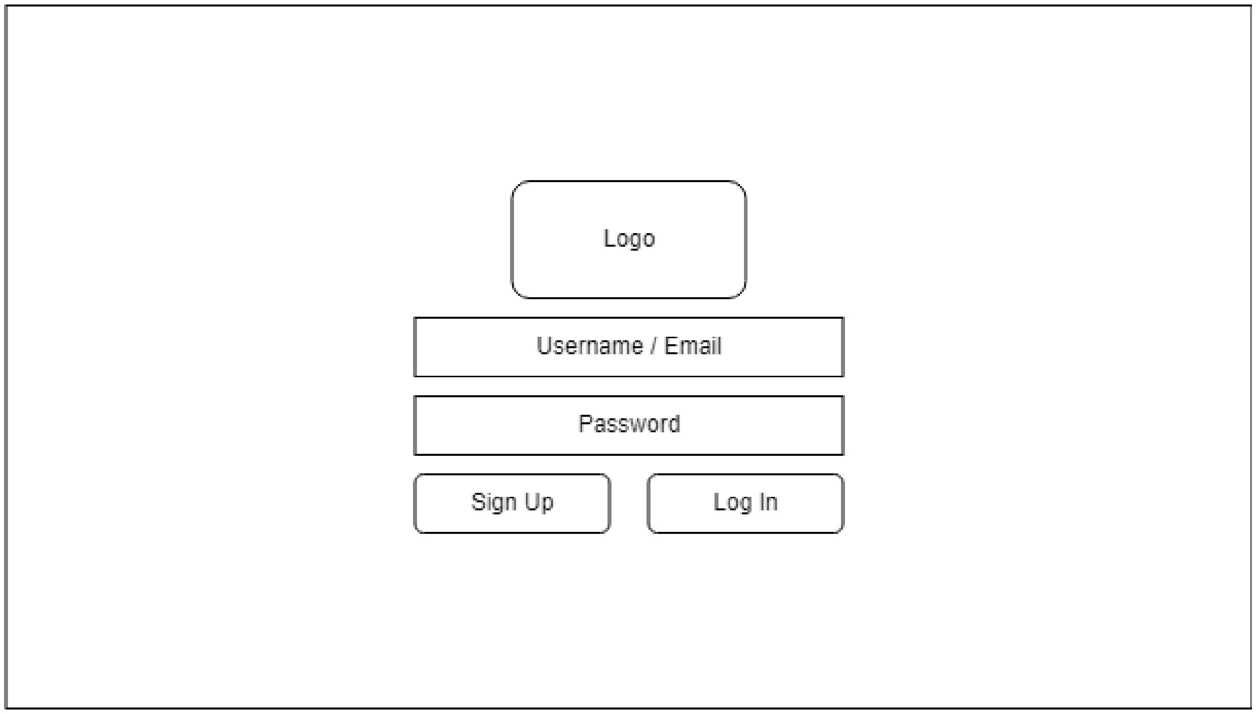

Figure 1.1 – Login Page

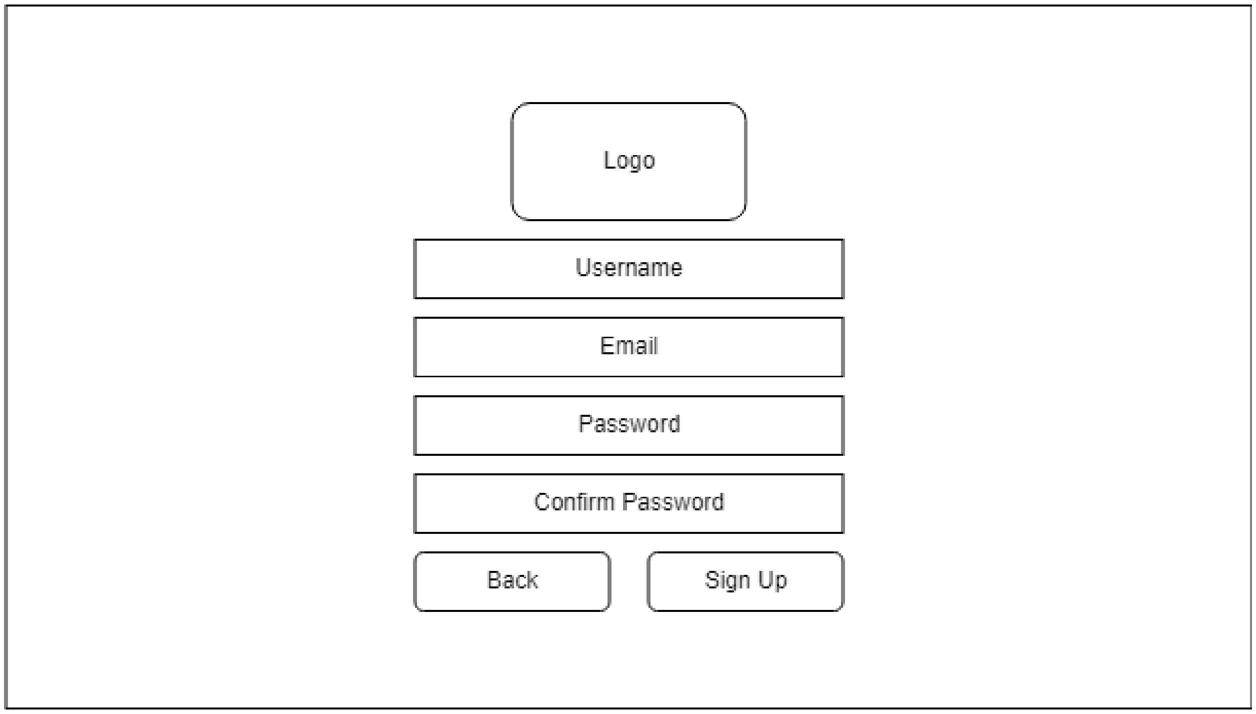

Figure 1.2 – Registration Page

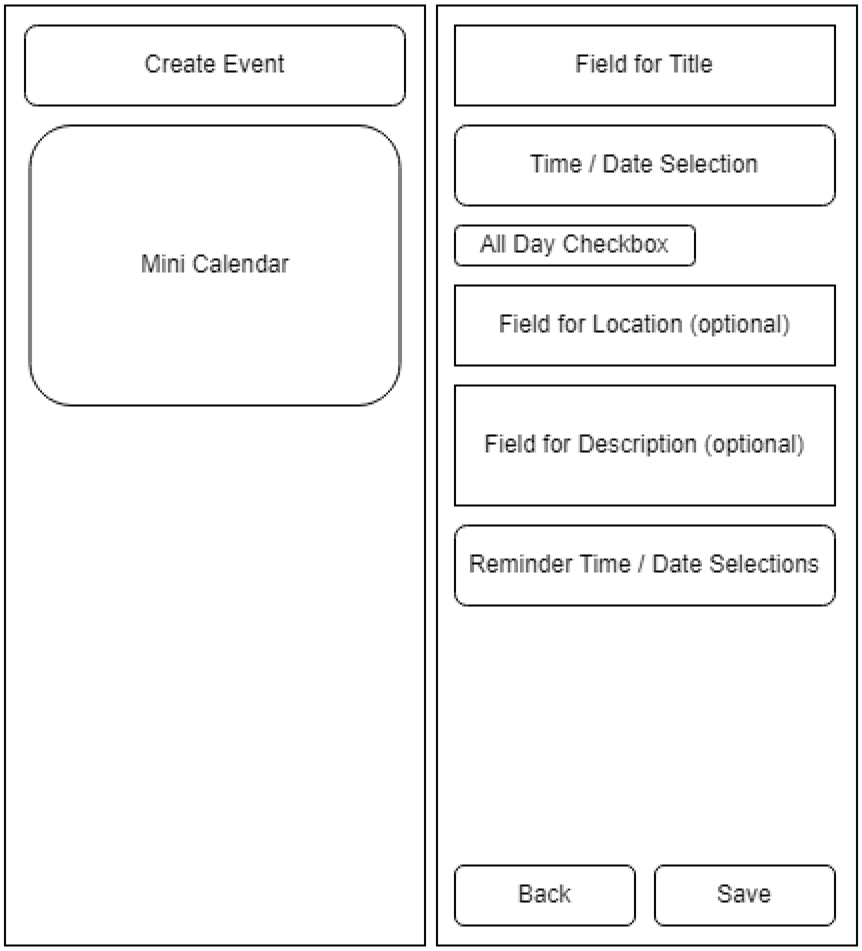
Create Event
Mini Calendar
Field for Title
Time / Date Selection
All Day Checkbox
Field for Location (optional)
Field for Description (optional)
Reminder Time / Date Selections
Back
Save

Figure 1.3 – Main Sidebar | *Figure 1.4 – Event Creation Sidebar*

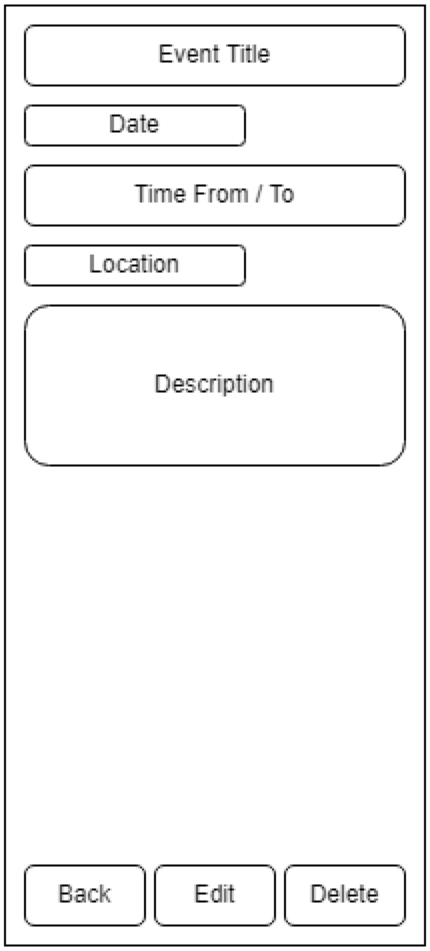

Figure 1.5 – Event Information Sidebar

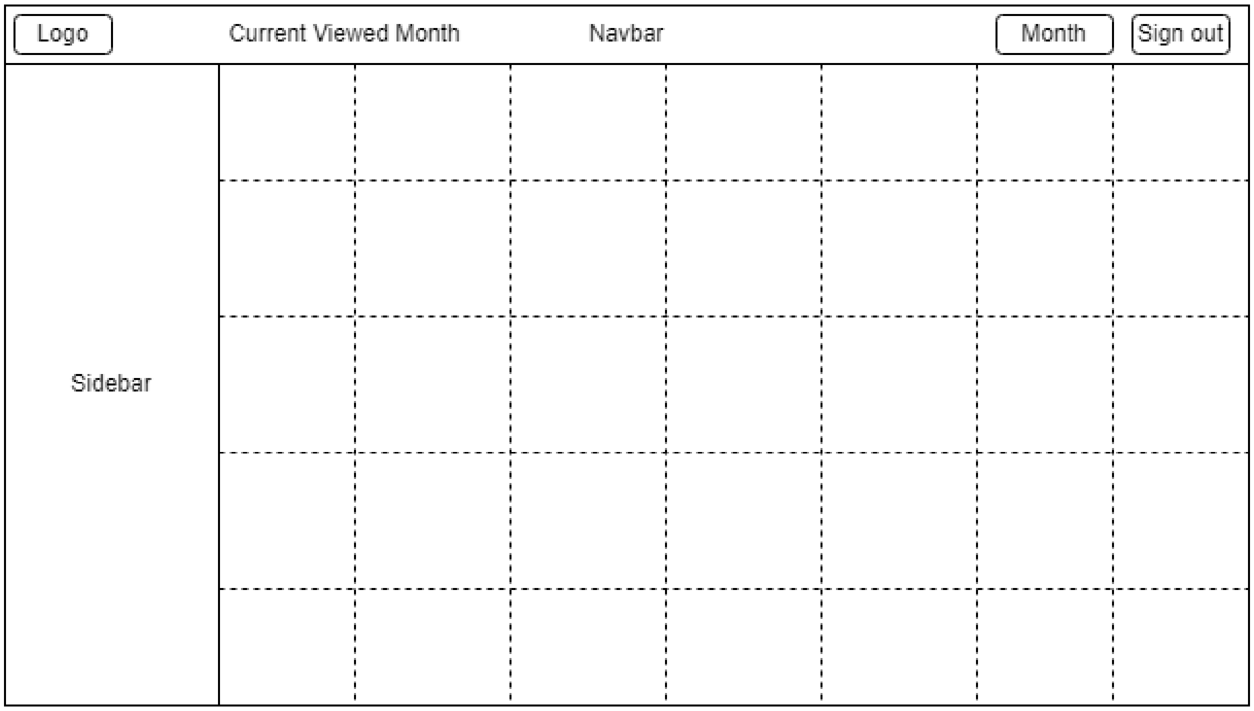

Figure 1.6 – Month View

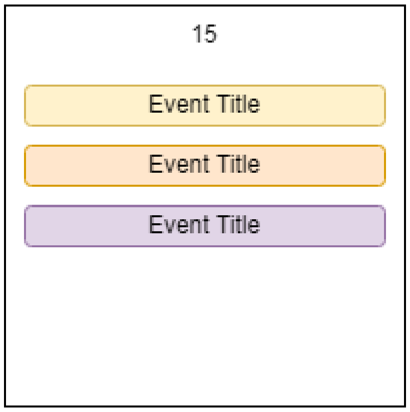

Figure 1.7 – Zoomed In Month View Cell

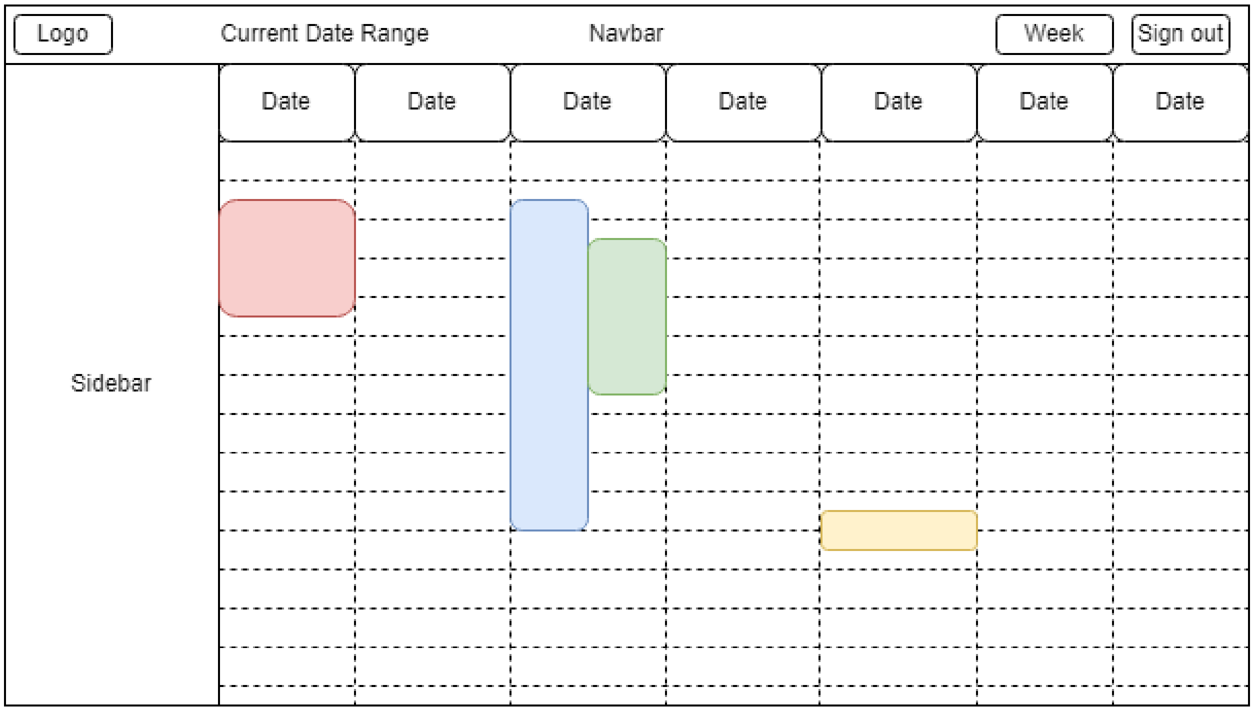

Figure 1.8 – Week View

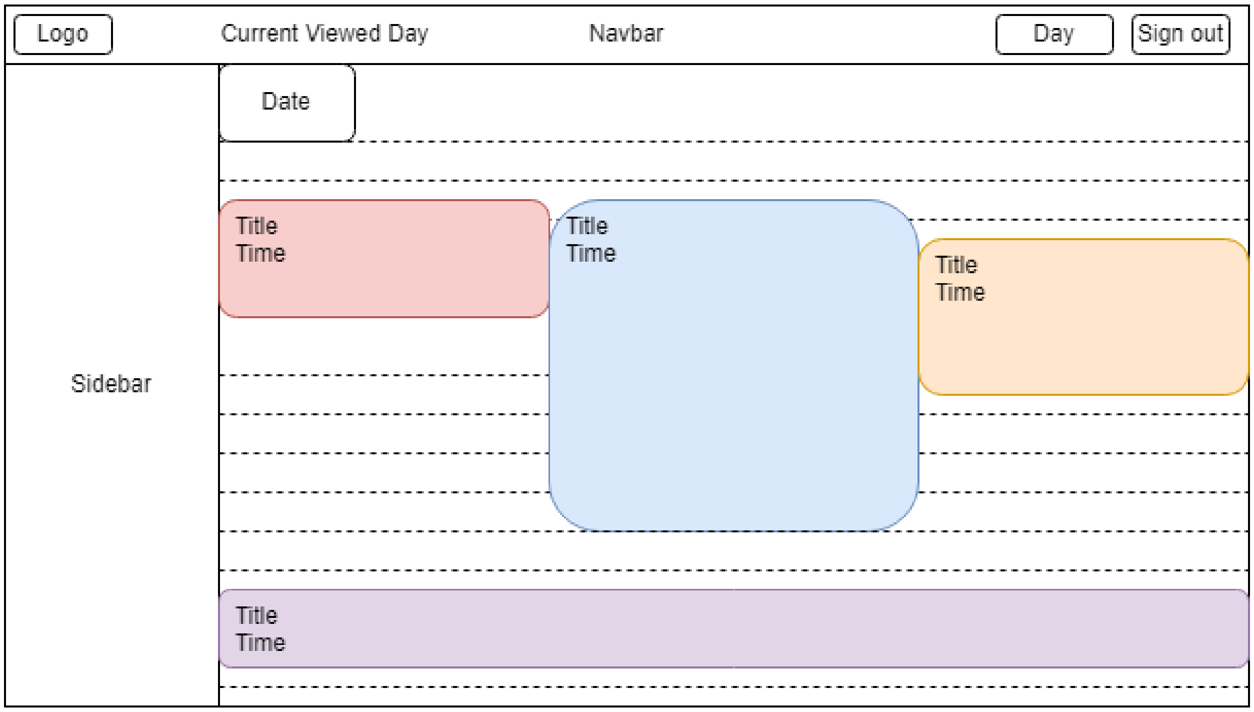

Figure 1.9 – Day View

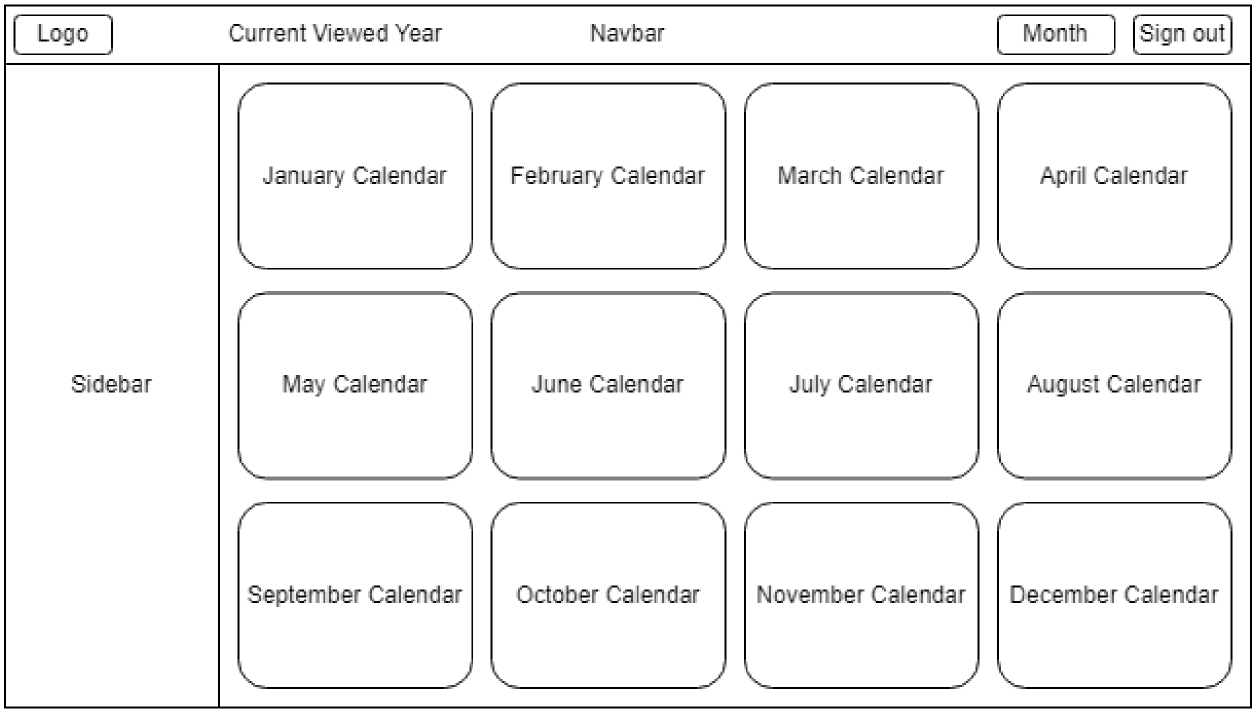

Figure 1.10 – Year View

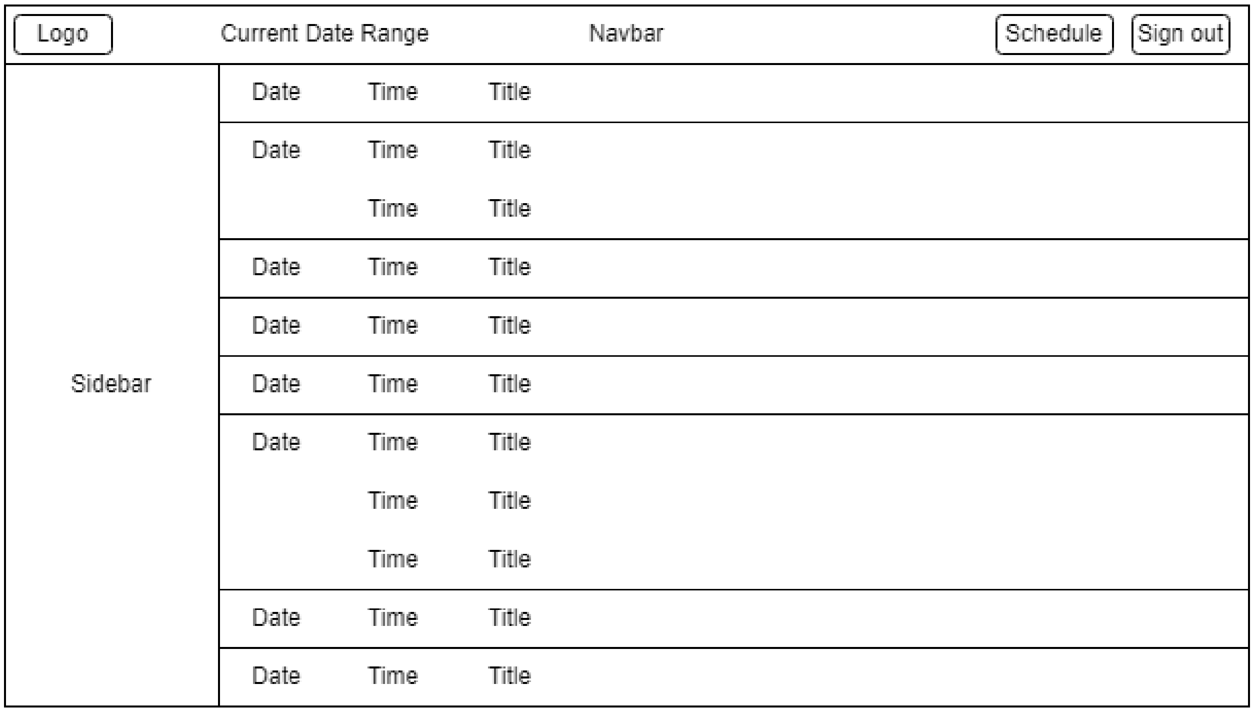

Figure 1.11 – Schedule View