What is Our Birthday Reminder App?

Imagine you have lots of friends, and you want to remember their birthdays so you can celebrate with them! This app helps you do just that. You can add their names and birthdays, see them all in one place, and even take them off if you need to.

Agenda

Topics Covered

1. Introduction
2. App Overview
3. User Interface
4. How It Works
5. Technology Used
6. Fun Features and Enhancements
7. Learning Opportunities
8. Conclusion

Introduction

Brief overview of the project
Importance of remembering birthdays

Go Back to Agenda Page

App Overview

What the app does

Main features: adding, viewing, and removing birthdays

Go Back to Agenda Page

Create the Main Components

App: The main component.

BirthdayForm: A form to add birthdays.

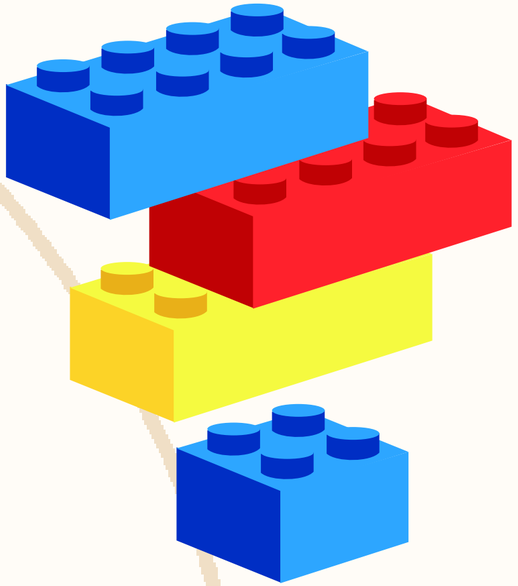
BirthdayList: A list to display the birthdays.

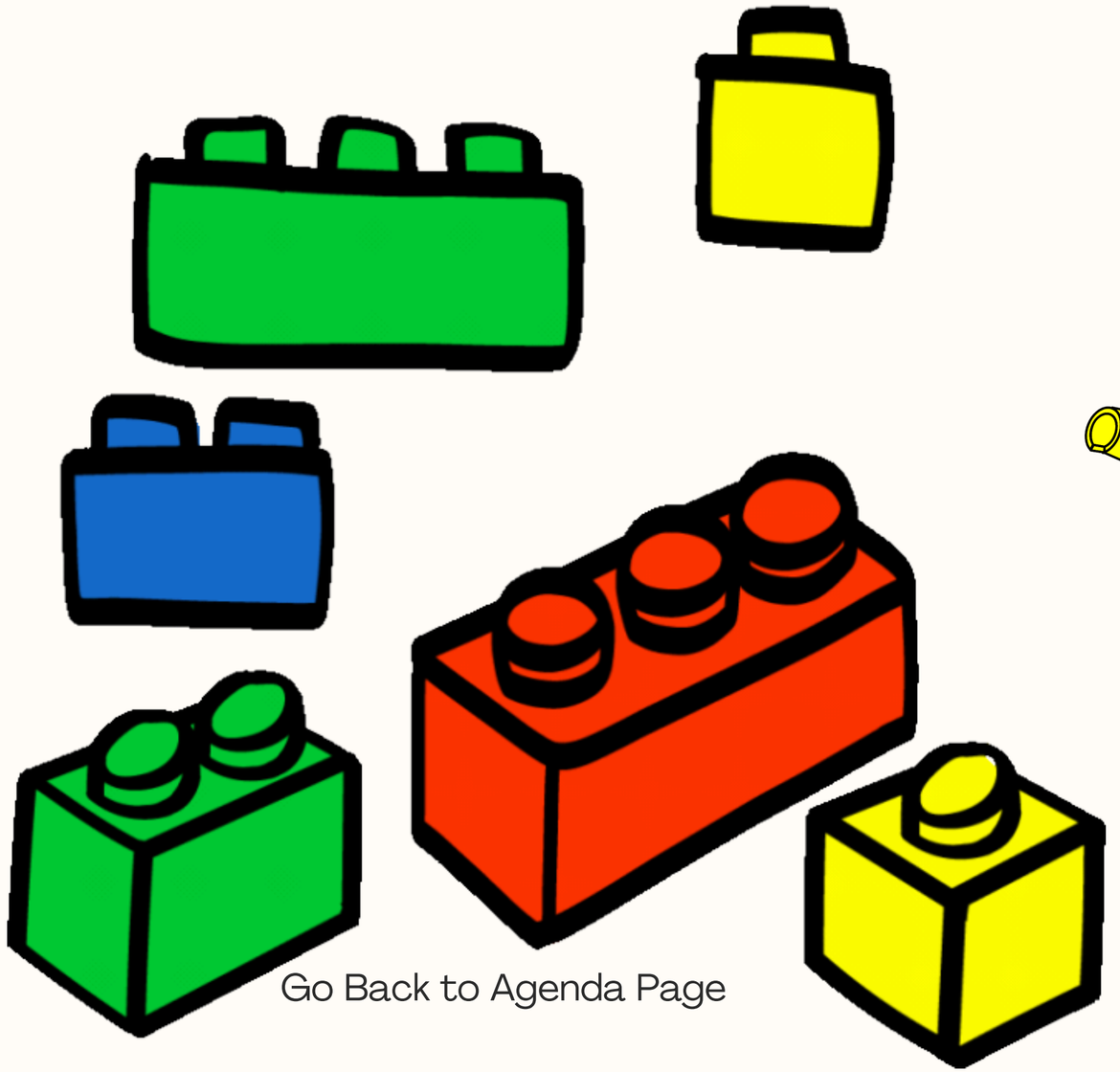
Go Back to Agenda Page

Components

Think of a component like a LEGO block. Just like how you can use LEGO blocks to build different things, in React, components are the building blocks we use to create our app!.

Go Back to Agenda Page





Go Back to Agenda Page

What is App.js?

Think of App.js as the main part of our birthday reminder app.

It’s like the boss that helps everything work together!

Here’s how it does that:

Go Back to Agenda Page

1. Starting Up the App

When we open our app, the first thing that happens is that it sets up a space to keep track of all the birthdays. We call this space state.

```
const [birthdays, setBirthdays] = useState([]);
```

What Does This Mean?

birthdays: This is where we keep all the names and dates of our friends' birthdays.
setBirthdays: This is like a tool that lets us change or update our birthday list.

2. Adding Birthdays

When you want to remember a new birthday, you use a special function called addBirthday.

```
const addBirthday = (birthday) => {
  setBirthdays([...birthdays, birthday]);
};
```

What Does This Mean?

When you give it a birthday (like a name and a date), it takes the old list (birthdays) and adds the new birthday to it.

The ...birthdays part is like saying, “I want to keep all the old birthdays and just add this new one.”

3. Removing Birthdays

If you want to take a birthday off the list, we have another function called removeBirthday.

```
const removeBirthday = (index) => {
  const newBirthdays = birthdays.filter((_, i) => i
!== index);
  setBirthdays(newBirthdays);
};
```

What Does This Mean?

- This function uses something called index to know which birthday to remove.
- filter is like a magic spell that helps us create a new list without the birthday we don’t want anymore.
- Finally, it updates our list with the new birthdays using setBirthdays.

4. Putting It All Together

The return part of App.js is where we put everything on the screen so you can see it.

```
return (
  <div style={{ textAlign: 'center', marginTop: '50px' }}>
    <h1>Birthday Reminder</h1>
    <BirthdayForm addBirthday={addBirthday} />
    <BirthdayList birthdays={birthdays} removeBirthday={removeBirthday} />
  </div>
);
```

What Does This Mean?

- The is like a box that holds everything.
- <h1> is a big title that says "Birthday Reminder."
- <BirthdayForm> is where you can add new birthdays. It uses the addBirthday function we made.
- <BirthdayList> is where you can see all the birthdays. It uses the birthdays list and the removeBirthday function to manage them.

Summary

App.js is the main part of our app that helps us remember birthdays.
It keeps track of birthdays, adds new ones, and removes old ones.

Finally, it shows everything on the screen so we can see our birthday list!

This is how App.js works to keep our birthday reminder app running smoothly! If you have any questions, feel free to ask! 🎉