

1. Import necessary libraries and components

- React, useState, useRef
- View, Text, TextInput, StyleSheet, TouchableOpacity, ScrollView, Alert, ImageBackground
- ViewShot for capturing views as images
- FileSystem and Sharing for saving and sharing files
- StatusBar for handling the device's status bar styling

2. Define main App component

- Initialize state variables: recipientName (text), message (text), decoration (emoji), cardColor (string for hex color)
- Create a ref (viewShotRef) for capturing the card component

3. Define handleCardCreation function

- Validate recipientName; show alert if empty
- Use ViewShot's `capture` method to capture the card
- Save the image to a file using FileSystem's moveAsync
- Share the saved file using Sharing API
- Use try-catch to handle and log errors

4. Define handleCardClick function

- Create an array of vibrant colors
- Select a random color using `Math.random` and update the cardColor state

5. Return JSX structure

- Use `ImageBackground` for the app's festive theme
- Use `ScrollView` to make content scrollable

- Include:
 - A header with the app's title
 - InputField components for recipient name and message
 - DecorationSelector for selecting emoji
 - Card component for previewing the birthday card
 - Button to create and download the card

6. Define reusable components:

- **InputField**: Customizable TextInput component with label and validation
- **DecorationSelector**: Renders decoration options as buttons, triggers a callback on selection
- **Card**: Displays the preview with the current decoration, recipient name, and message

7. Style components using StyleSheet

- **Background**: Full-screen image with flex layout
- **Header**: Bold text, centered, festive color
- **Card**: Dynamic background color, padding, and rounded corners
- **Button**: Vibrant color, bold text, and padding
- Ensure responsiveness for various screen sizes

8. Test the app:

- Test on physical devices and simulators for iOS and Android
- Validate ViewShot and Sharing functionality
- Check UI responsiveness and user experience
- Ensure error handling is robust