**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**KHOA KHOA HỌC VÀ KĨ THUẬT THÔNG TIN**



**IE307.P11.VB2. CÔNG NGHỆ LẬP TRÌNH ĐA NỀN TẢNG CHO THIẾT BỊ DI ĐỘNG**

**Seminar – Nhóm 5: ListView Components - GridView Components - Expandable - Collapsible Components**

**GVHD:** ThS. Võ Ngọc Tân

**SVTH:**

1. Phạm Minh Quốc MSSV: 22540017

🙡🙢 Tp. Hồ Chí Minh, 10/2024 🙠🙣

**NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN**

*……., ngày……...tháng……năm 20…*

**Người nhận xét**

*(Ký tên và ghi rõ họ tên****)***

MỤC LỤC

[Chương 1: ListView Components 5](#_Toc181442609)

[1.1 SectionList: 5](#_Toc181442610)

[1.1.1 Cách hoạt động: 5](#_Toc181442611)

[1.1.2 Cách hiện thực hóa: 5](#_Toc181442612)

[1.1.3 Mã nguồn ví dụ: 6](#_Toc181442613)

[1.1.4 Hình ảnh kết quả: 7](#_Toc181442614)

[1.2 Expandable List View: 7](#_Toc181442615)

[1.2.1 Cách hoạt động: 7](#_Toc181442616)

[1.2.2 Mã nguồn ví dụ: 8](#_Toc181442617)

[1.2.3 Hình ảnh kết quả: 10](#_Toc181442618)

[1.3 Show Timeline ListView: 10](#_Toc181442619)

[1.3.1 Cấu trúc: 10](#_Toc181442620)

[1.3.2 Mã nguồn ví dụ: 10](#_Toc181442621)

[1.3.3 Hình ảnh kết quả: 25](#_Toc181442622)

[Chương 2: GridView Components 30](#_Toc181442623)

[2.1 GridView: 30](#_Toc181442624)

[2.1.1 Cấu trúc: 30](#_Toc181442625)

[2.1.2 Cách hiện thực hóa: 30](#_Toc181442626)

[2.1.3 Mã nguồn ví dụ: 31](#_Toc181442627)

[2.1.4 Hình ảnh kết quả: 32](#_Toc181442628)

[2.2 View Like Google Play Store: 32](#_Toc181442629)

[2.2.1 Cấu trúc: 32](#_Toc181442630)

[2.2.2 Cách hiện thực hóa: 32](#_Toc181442631)

[2.2.3 Mã nguồn ví dụ: 33](#_Toc181442632)

[2.2.4 Hình ảnh kết quả: 35](#_Toc181442633)

[Chương 3: Expandable - Collapsible Components 37](#_Toc181442634)

[3.1 Tree View: 37](#_Toc181442635)

[3.1.1 Cấu trúc và Tính năng: 37](#_Toc181442636)

[3.1.2 Cách hiện thực hóa: 37](#_Toc181442637)

[3.1.3 Mã nguồn ví dụ: 38](#_Toc181442638)

[3.1.4 Hình ảnh kết quả: 39](#_Toc181442639)

[3.2 Collapsible/Accordion: 39](#_Toc181442640)

[3.2.1 Cấu trúc và Tính năng: 40](#_Toc181442641)

[3.2.2 Cách hiện thực hóa: 40](#_Toc181442642)

[3.2.3 Mã nguồn ví dụ: 42](#_Toc181442643)

[3.2.4 Hình Ảnh kết quả: 45](#_Toc181442644)

# ListView Components

## SectionList:

Selection list, hay còn gọi là danh sách lựa chọn, là một thành phần giao diện người dùng quan trọng trong React Native, cho phép người dùng chọn một hoặc nhiều tùy chọn từ một danh sách. Thành phần này thường được sử dụng trong các ứng dụng di động để tạo trải nghiệm người dùng mượt mà và trực quan.

### Cách hoạt động:

Một selection list có thể được xây dựng bằng cách sử dụng các thành phần cơ bản như FlatList, Picker hoặc Modal. Khi người dùng nhấn vào một mục trong danh sách, lựa chọn của họ sẽ được lưu trữ và có thể được sử dụng để thực hiện các hành động khác nhau, chẳng hạn như gửi dữ liệu đến máy chủ hoặc thay đổi trạng thái giao diện.

### Cách hiện thực hóa:

Import thư viện:

*import {SectionList} from 'react-native'*

Rendering element:

*<SectionList*

*ItemSeparatorComponent={****Separator View****}*

*sections={****Section Data****}*

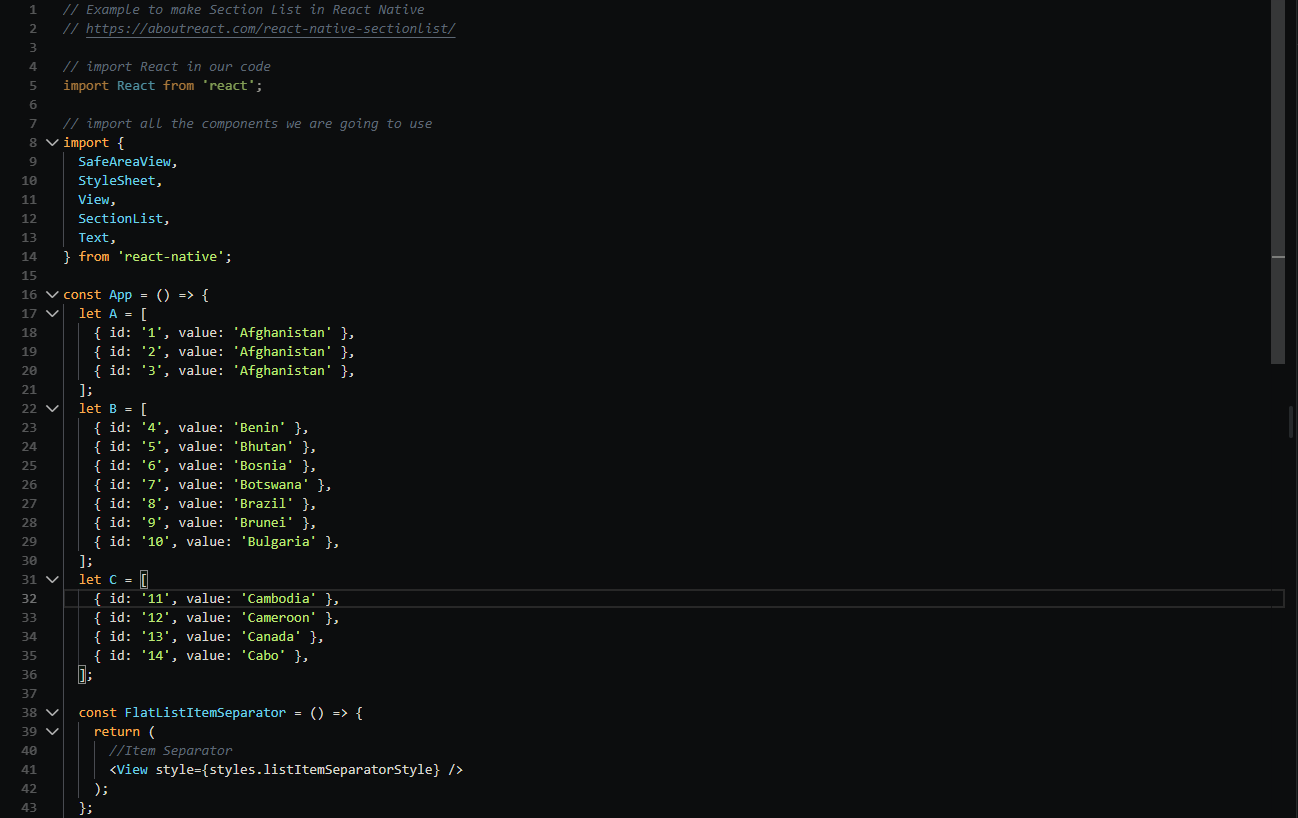
*renderSectionHeader={({ section }) => (****Header View****)}*

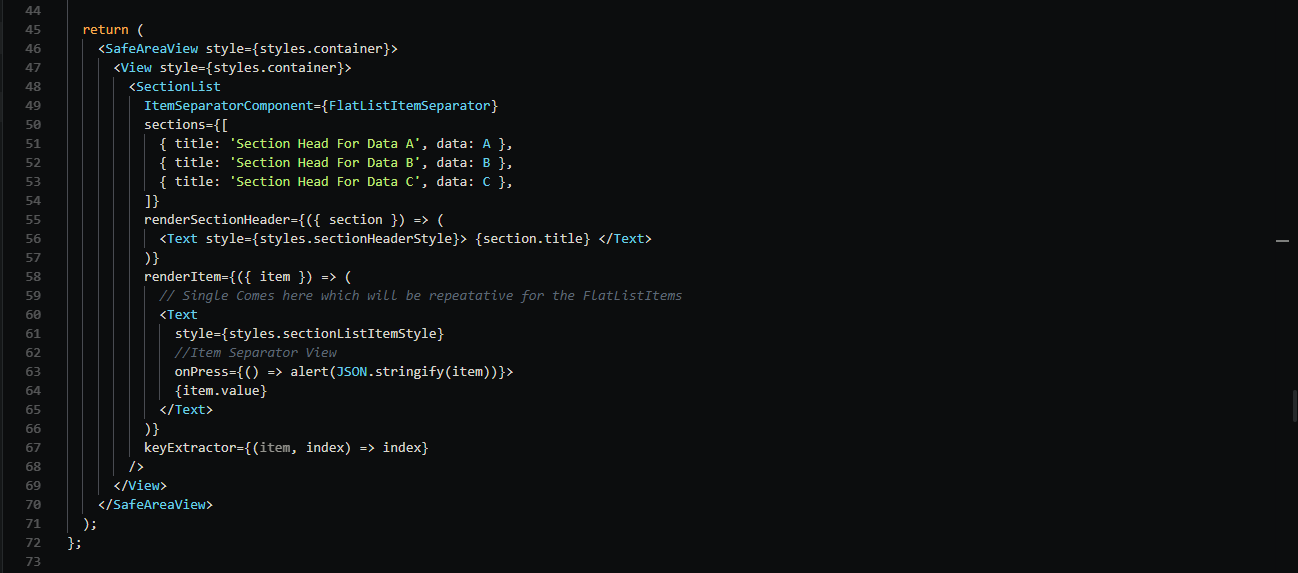
*renderItem={({ item }) => (****Single Item View****)}*

*keyExtractor={(item, index) => index}*

*/>*

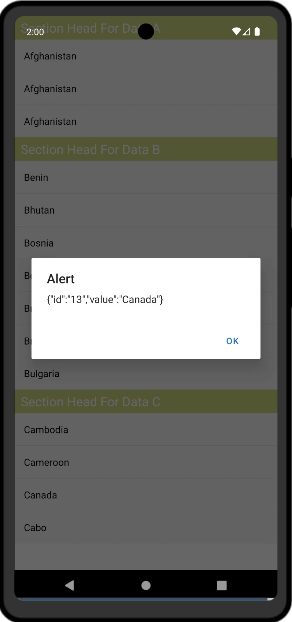
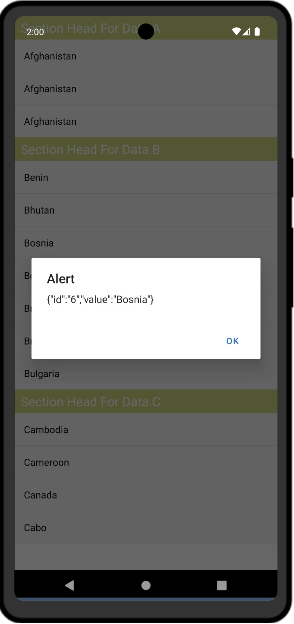
### Mã nguồn ví dụ:







### Hình ảnh kết quả:



## Expandable List View:

Expandable List View (Danh sách mở rộng) cho phép người dùng mở rộng và thu gọn các mục trong danh sách để xem thêm thông tin. Thành phần này rất hữu ích khi bạn có nhiều dữ liệu hoặc thông tin liên quan mà bạn muốn người dùng có thể truy cập mà không làm rối mắt giao diện.

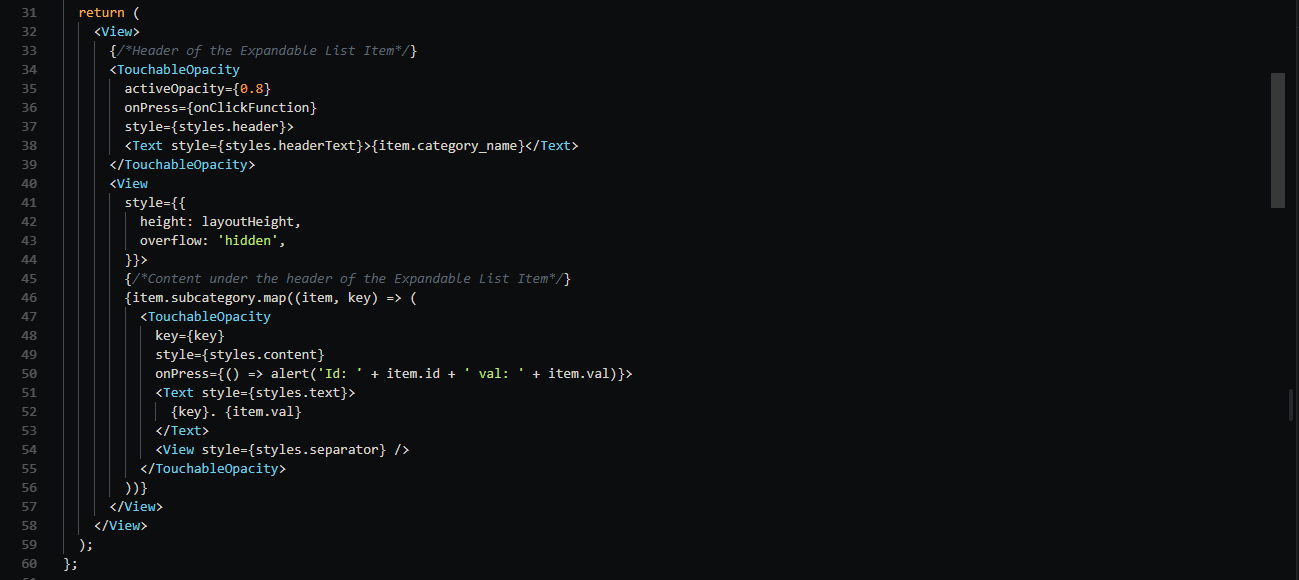
### Cách hoạt động:

Expandable List View thường bao gồm các mục có thể được mở rộng để hiển thị thêm nội dung khi người dùng nhấn vào chúng. Điều này giúp tiết kiệm không gian hiển thị và cho phép người dùng chỉ xem thông tin họ quan tâm.

Expandable ListView giống như Selection View nhưng có nội dung phần ẩn. Tiêu đề của danh sách sẽ luôn hiển thị và sau khi nhấp vào tiêu đề, bạn sẽ thấy tiêu đề phụ của listview

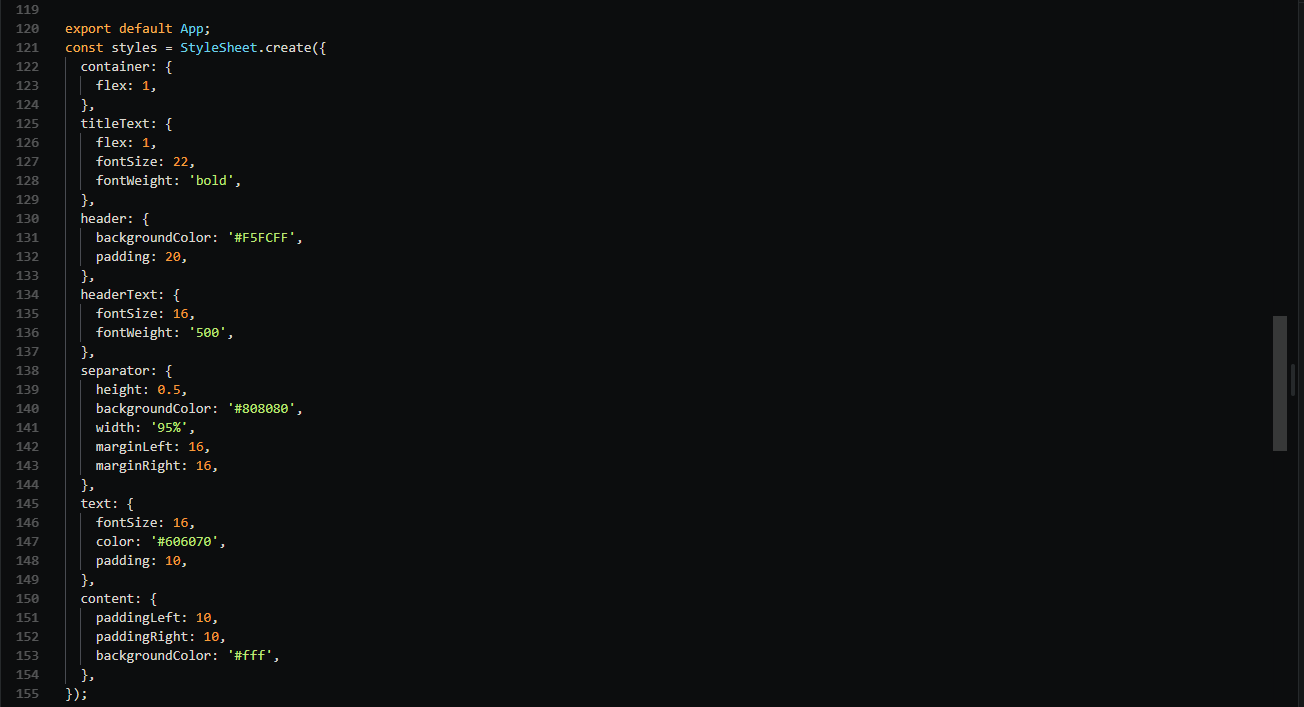
### Mã nguồn ví dụ:



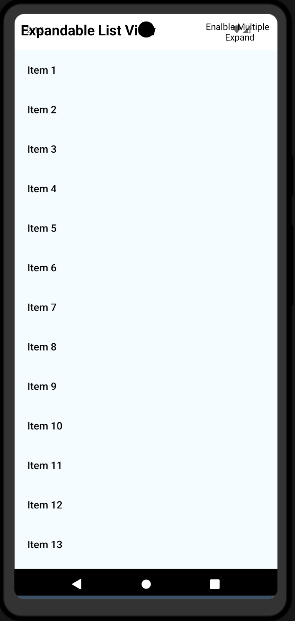
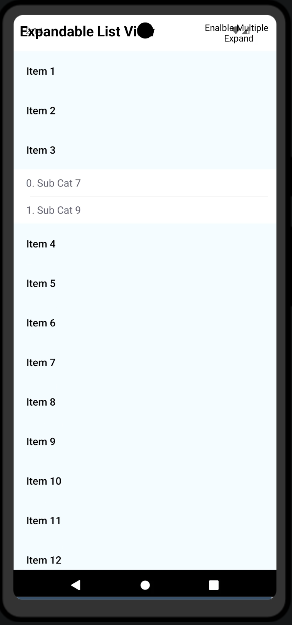








### Hình ảnh kết quả:

**** ****

## Show Timeline ListView:

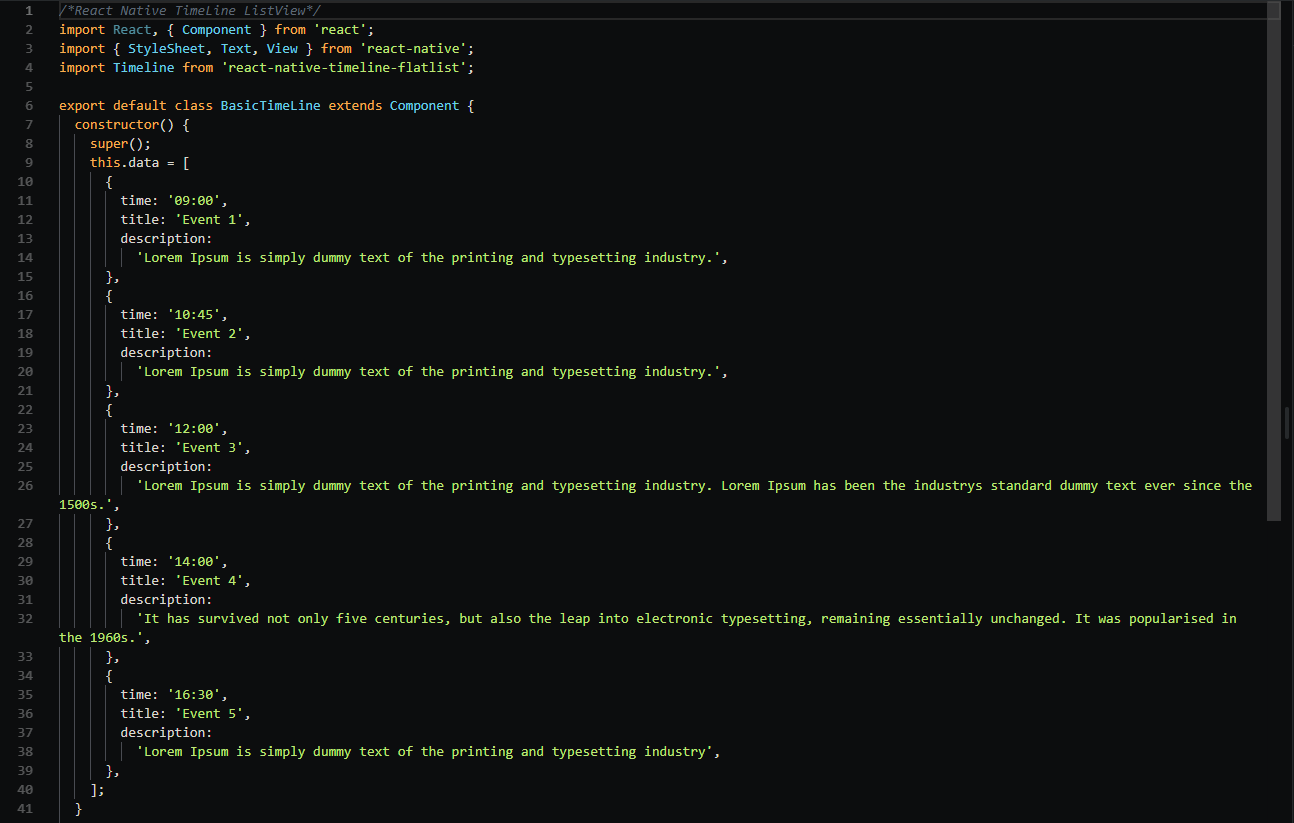
Timeline ListView (Danh sách theo dõi theo kiểu dòng thời gian) được sử dụng để hiển thị các sự kiện, hoạt động hoặc thông tin theo thứ tự thời gian. Thành phần này rất hữu ích trong các ứng dụng cần theo dõi lịch sử hoặc tiến trình, như ứng dụng quản lý dự án, nhật ký hoạt động, hoặc lịch sử giao dịch.

### Cấu trúc:

Timeline ListView thường bao gồm các mục với các thông tin như thời gian, tiêu đề và mô tả. Mỗi mục có thể được trình bày theo một cách trực quan, giúp người dùng dễ dàng nhận biết các sự kiện trong khoảng thời gian cụ thể.

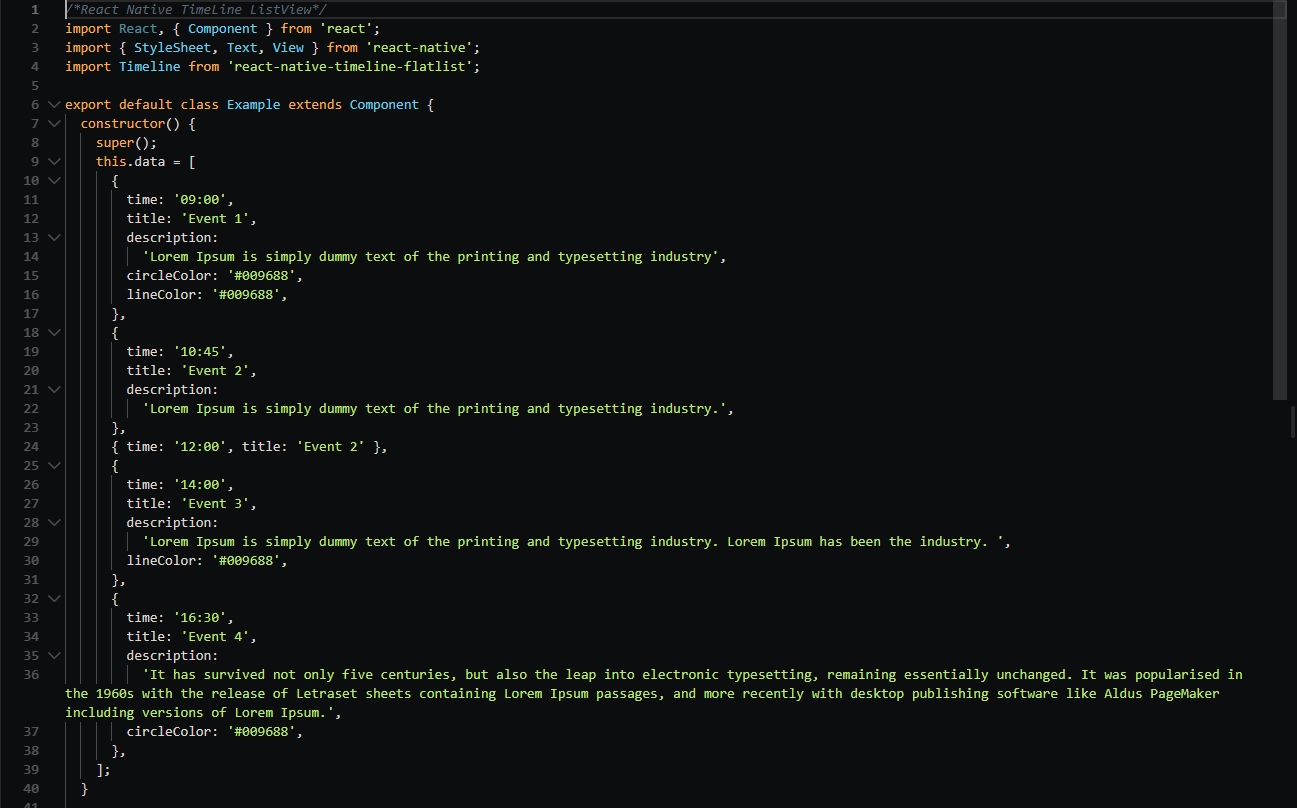
### Mã nguồn ví dụ:

***BasicTimeLine:***

****

****

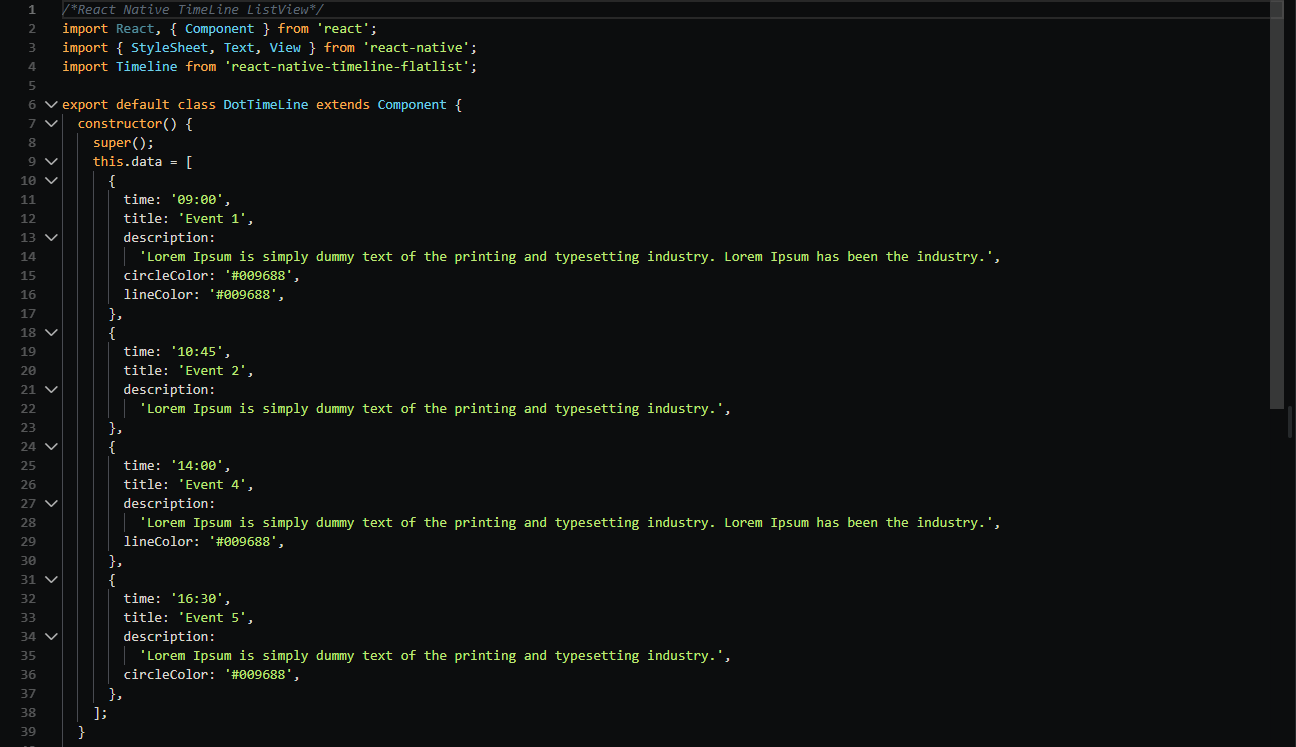
***CustomTimeLine:***

****

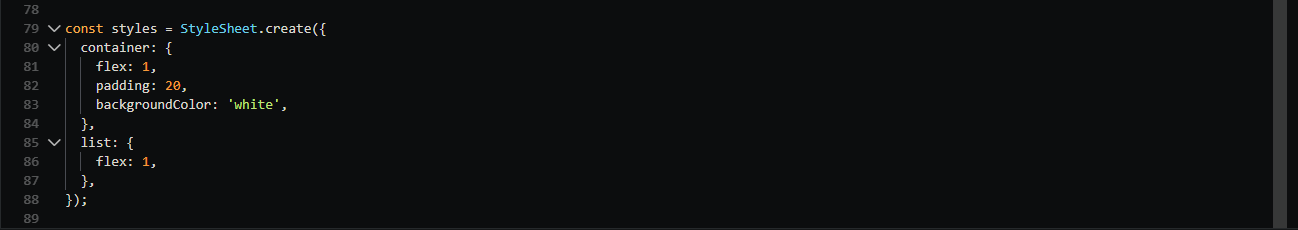
****

****

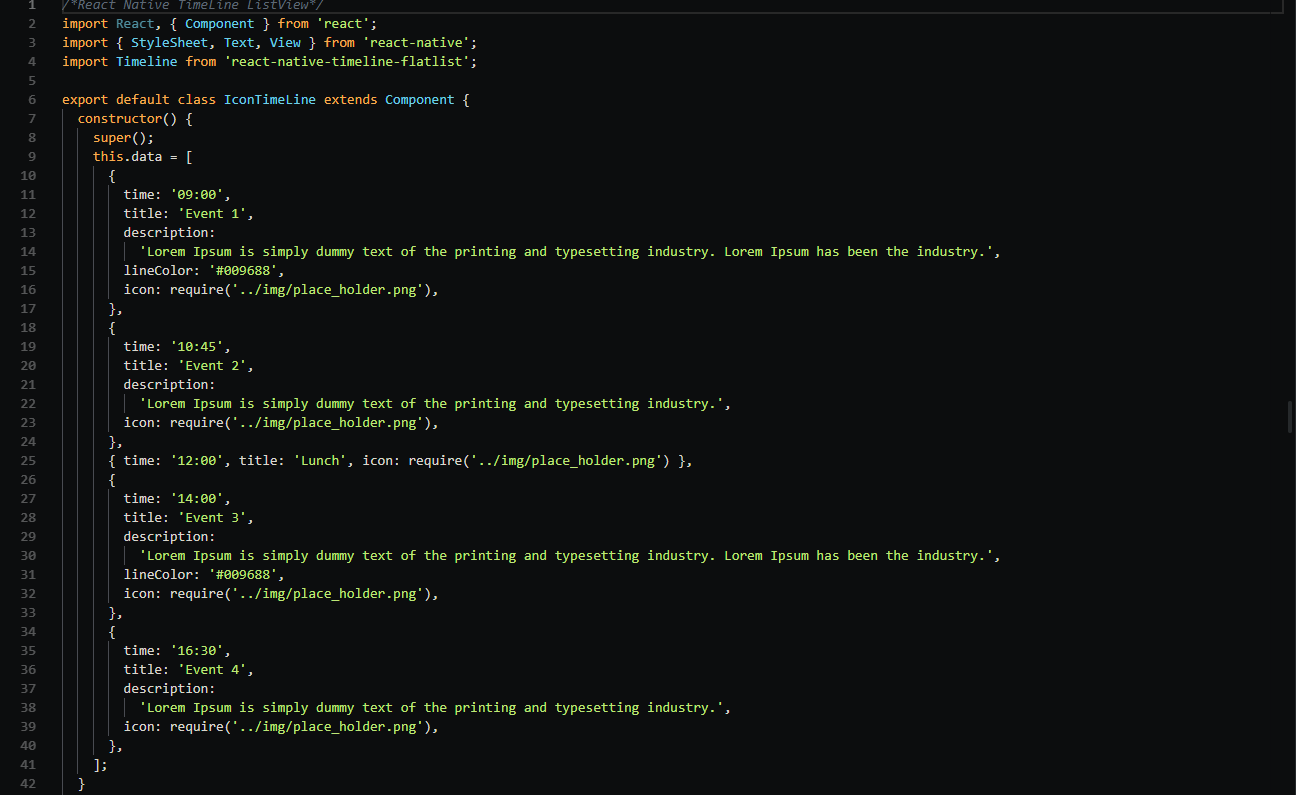
***DotTimeLine:***

******

******

******

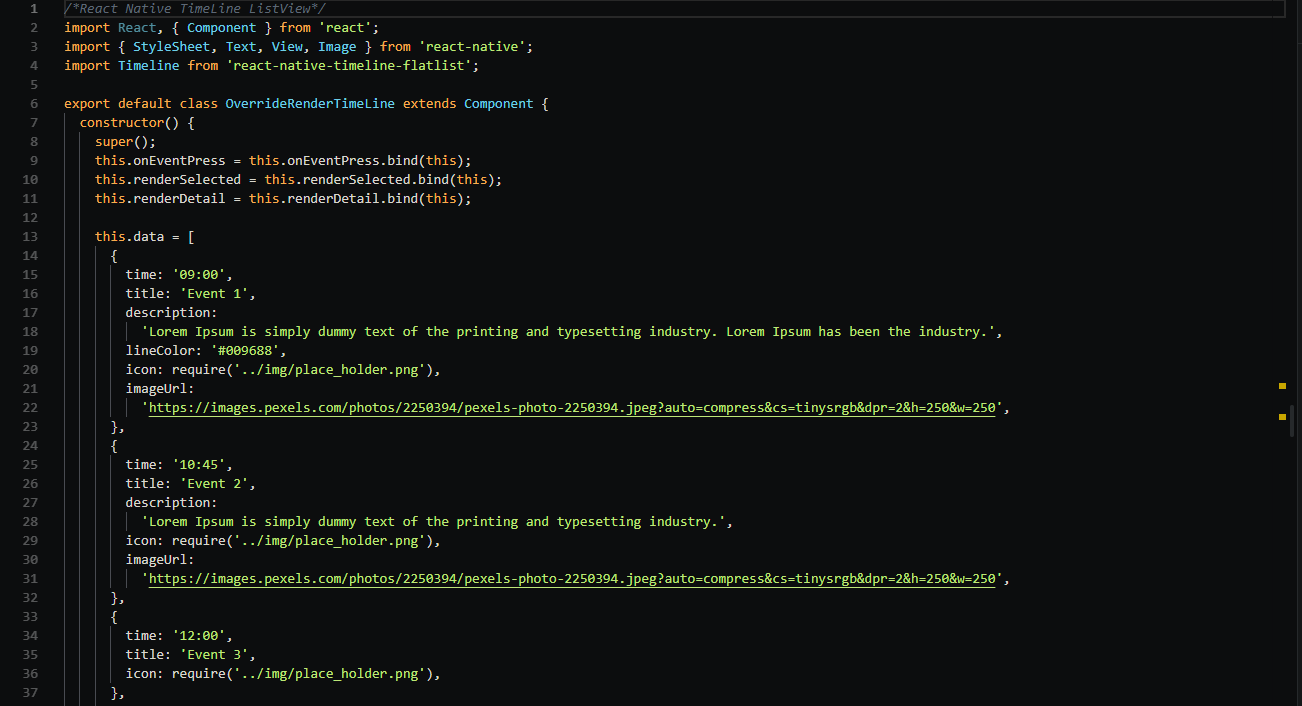
***IconTimeLine:***

******

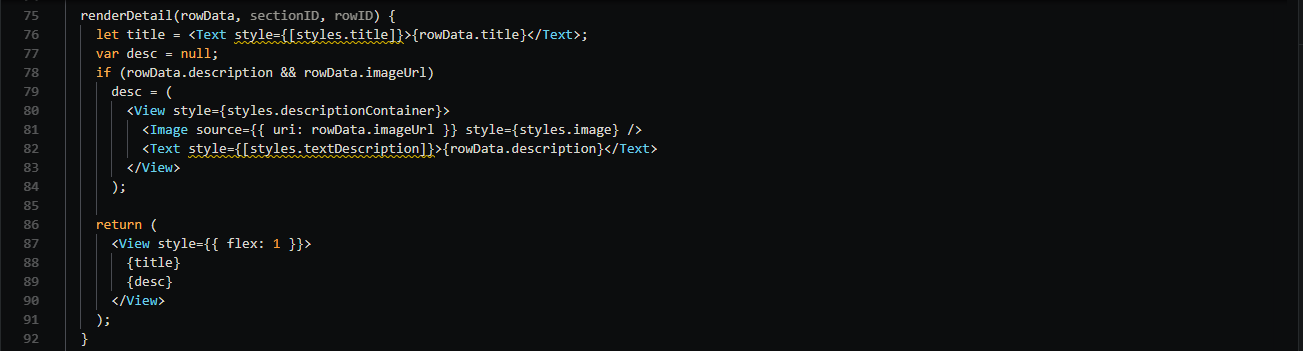
******

******

***OverrideRenderTimeLine:***

******

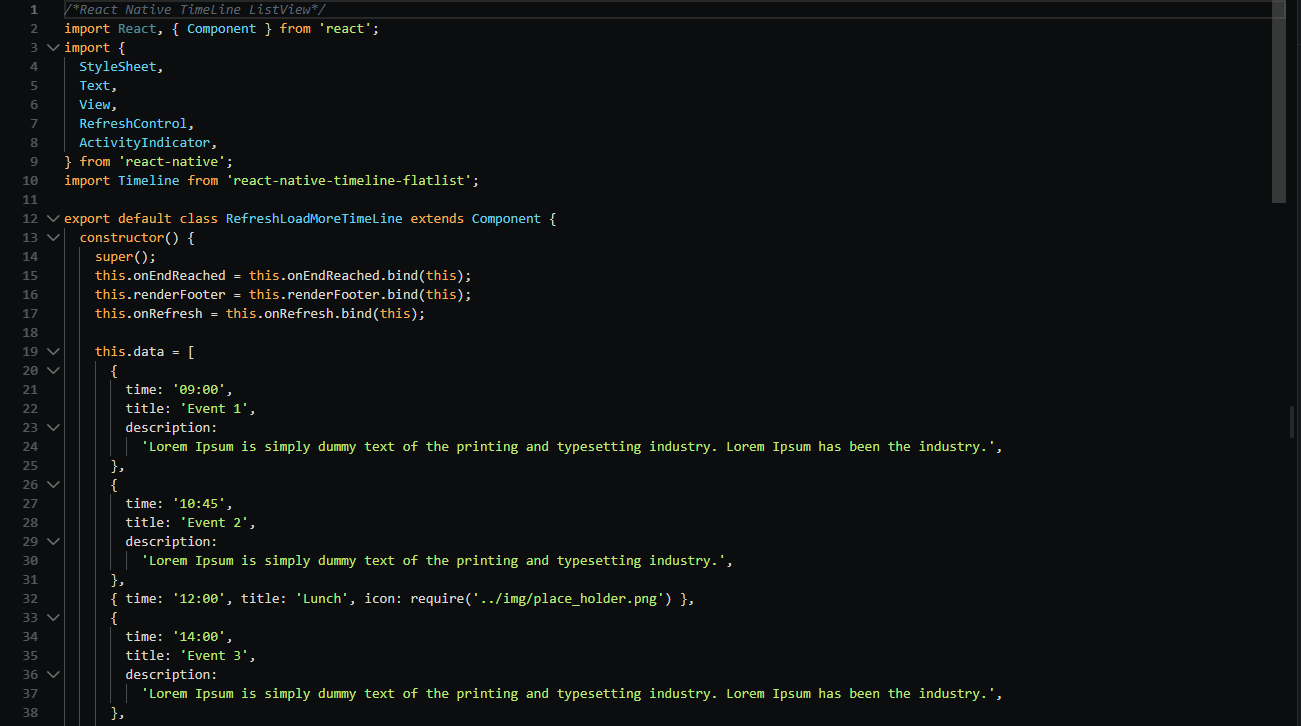
******

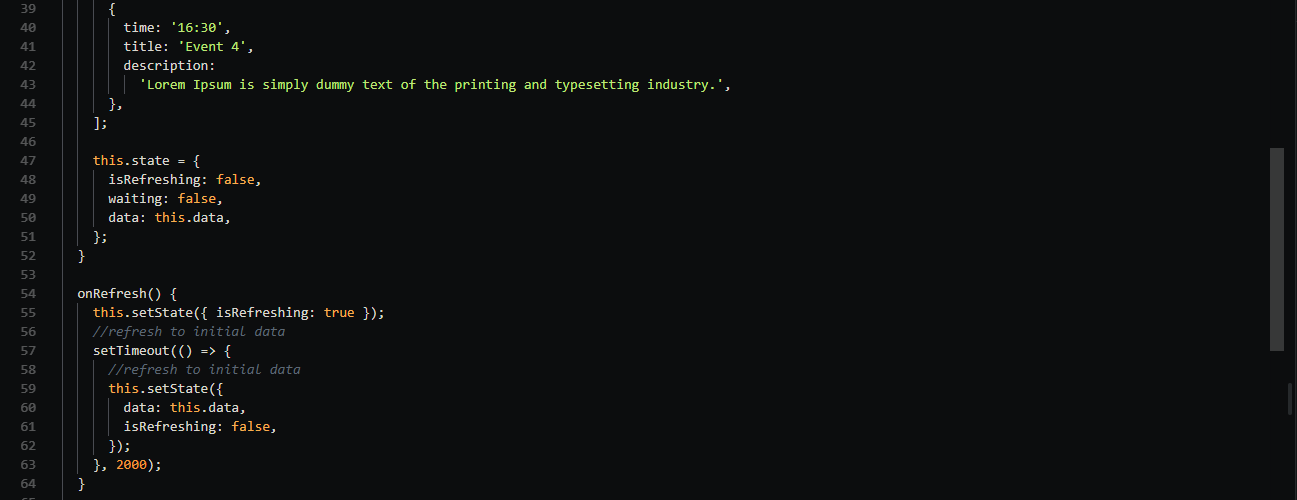
******

******

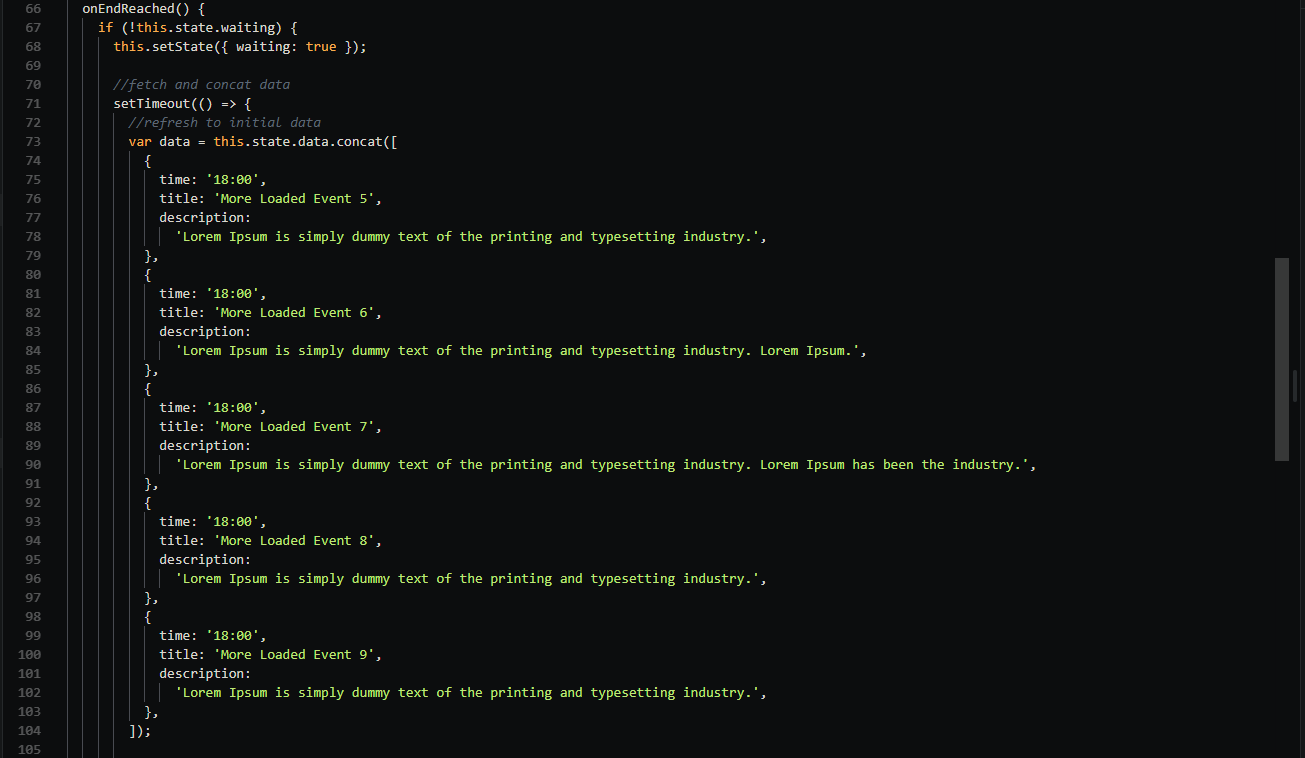
******

***RefreshLoadMoreTimeLine:***

******

******

******

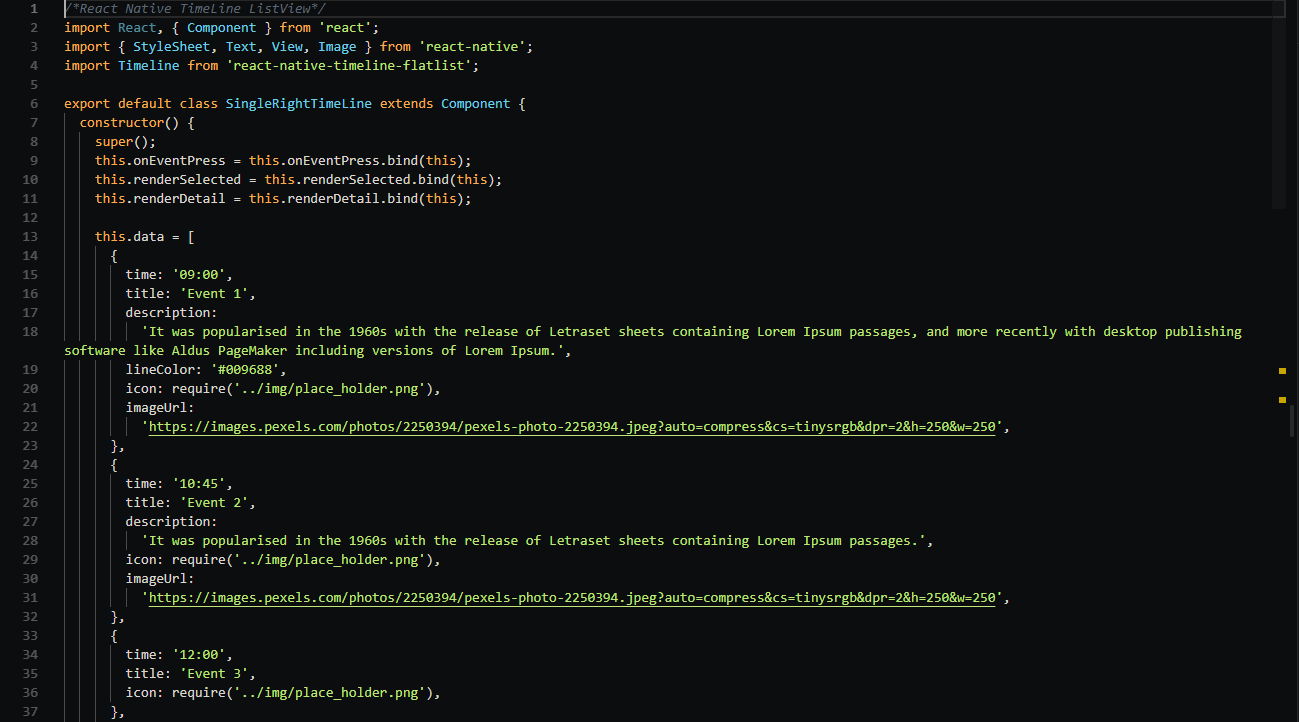
******

******

******

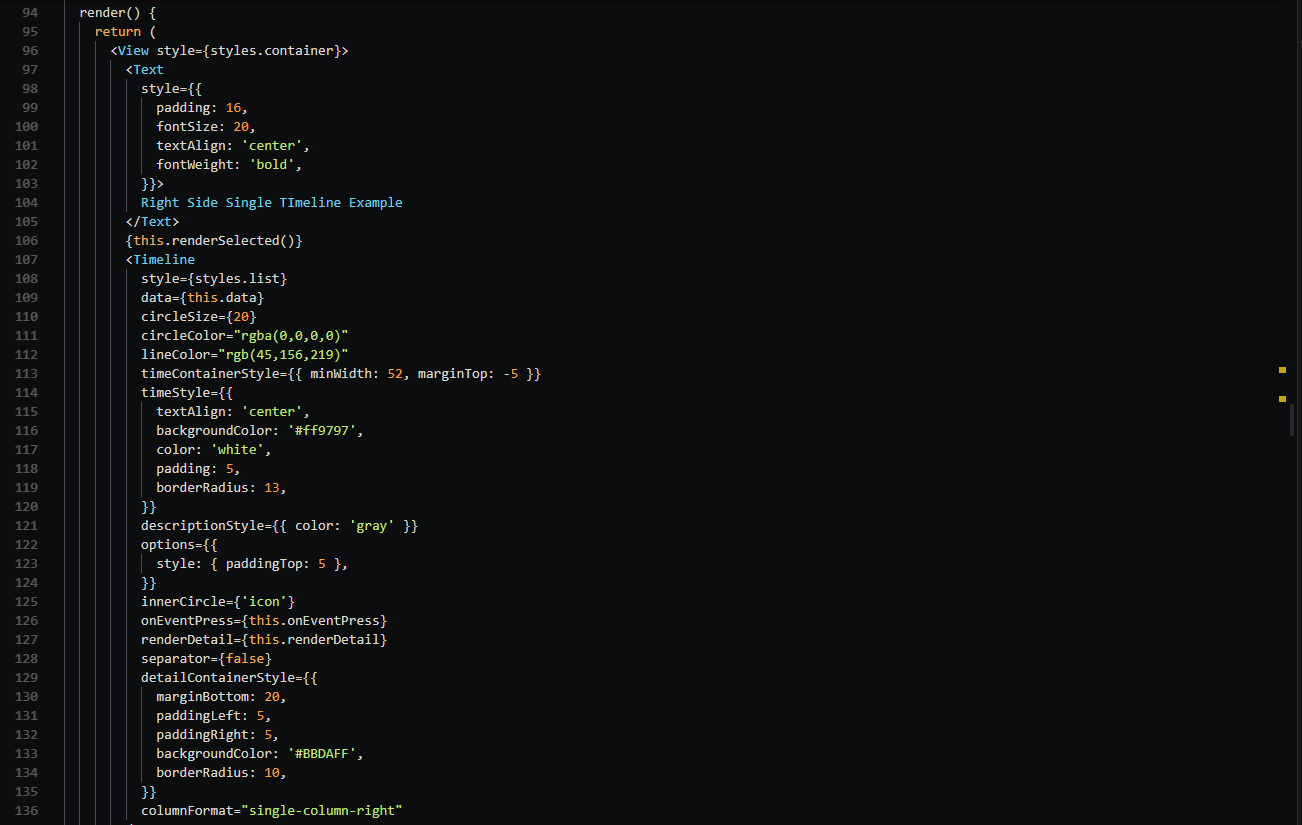
******

***SingleRightTimeLine:***

******

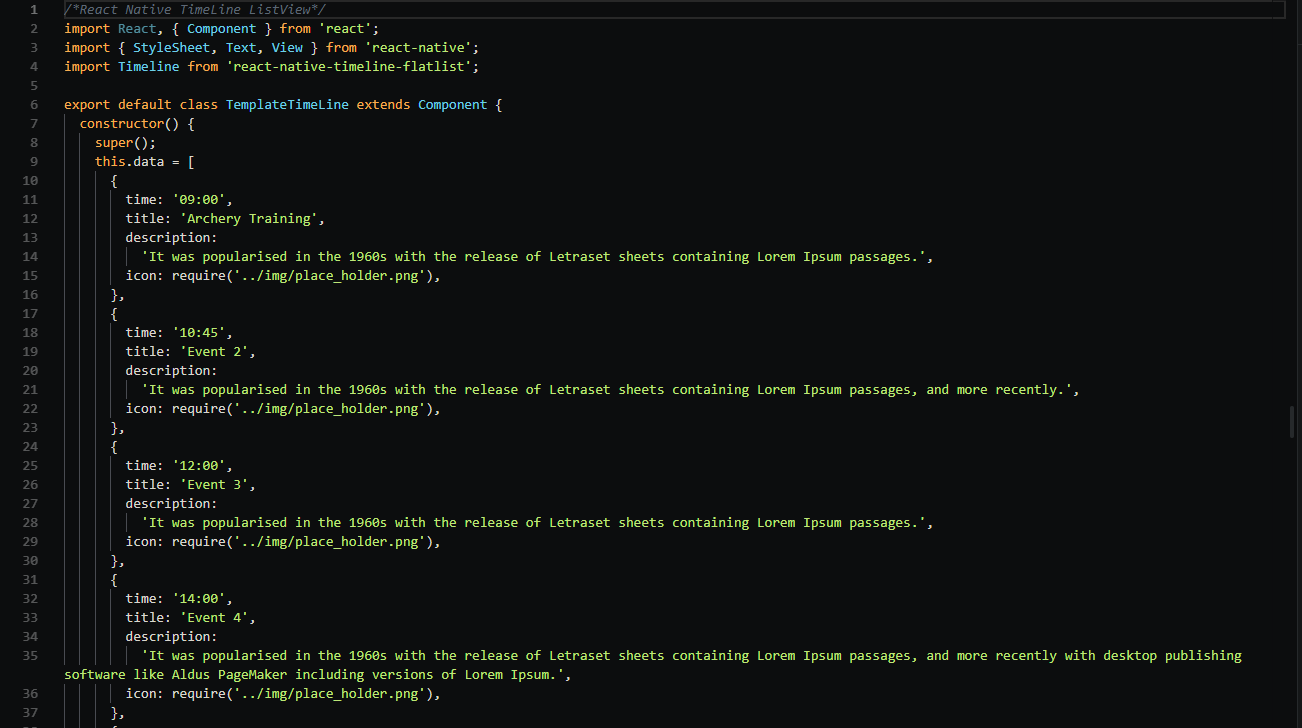
******

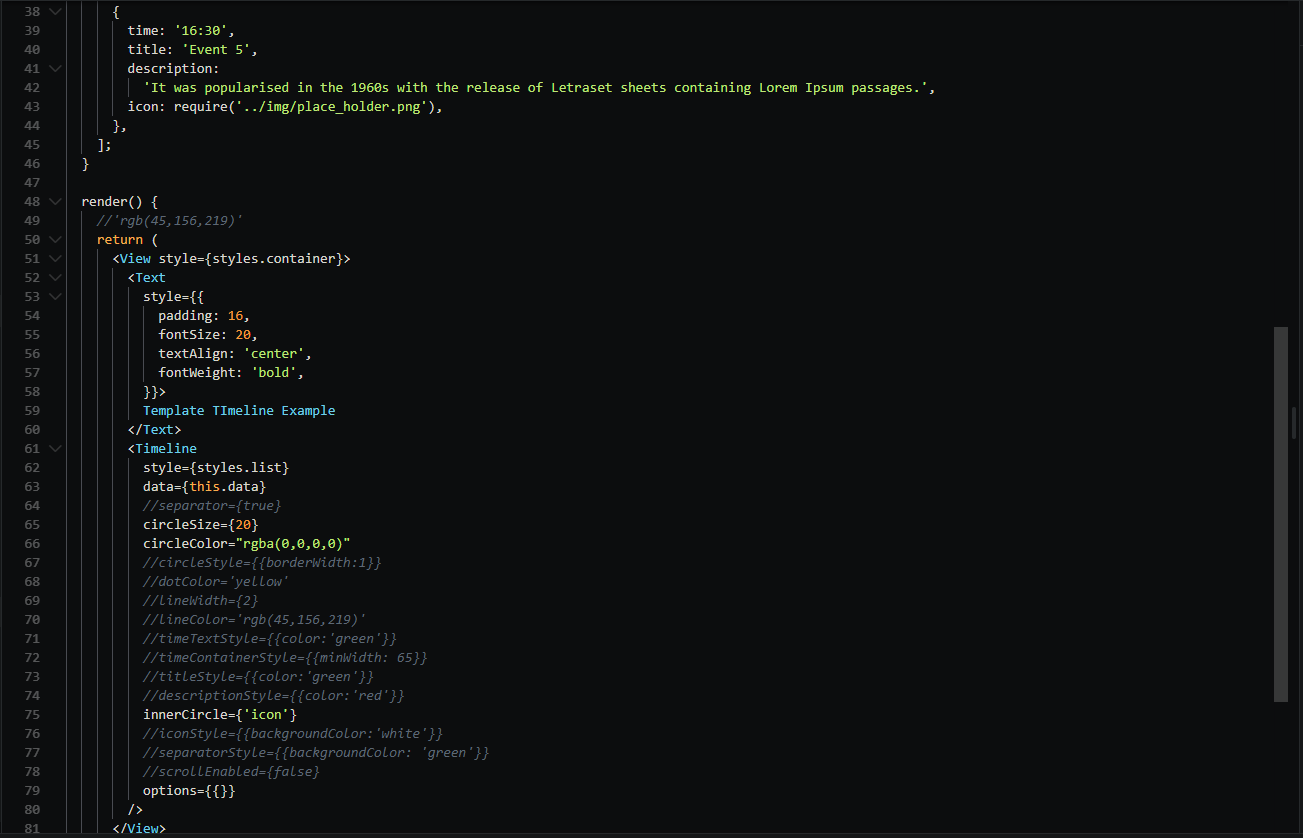
******

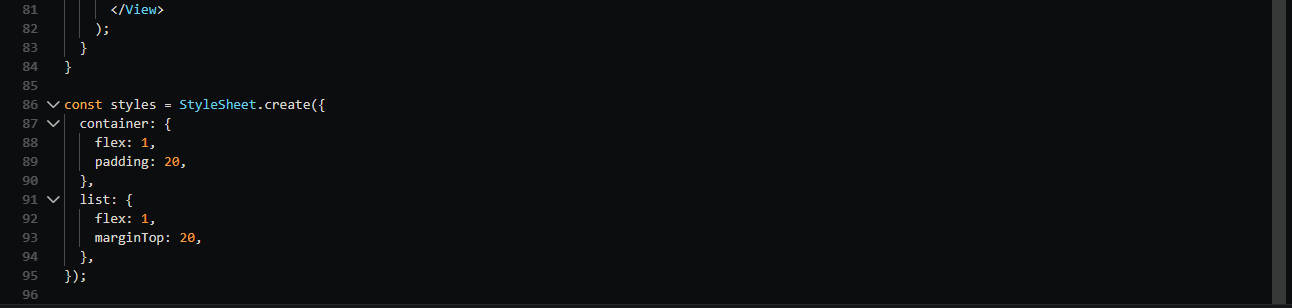
******

******

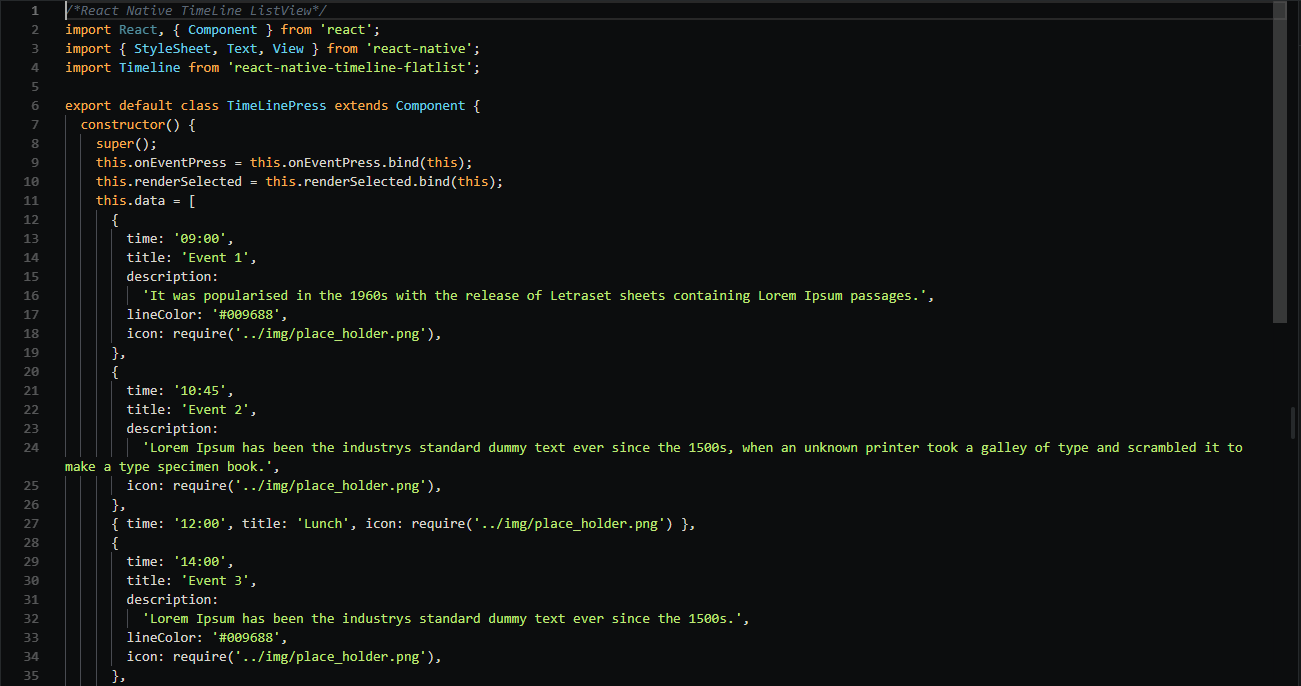
***TemplateTimeLine:***

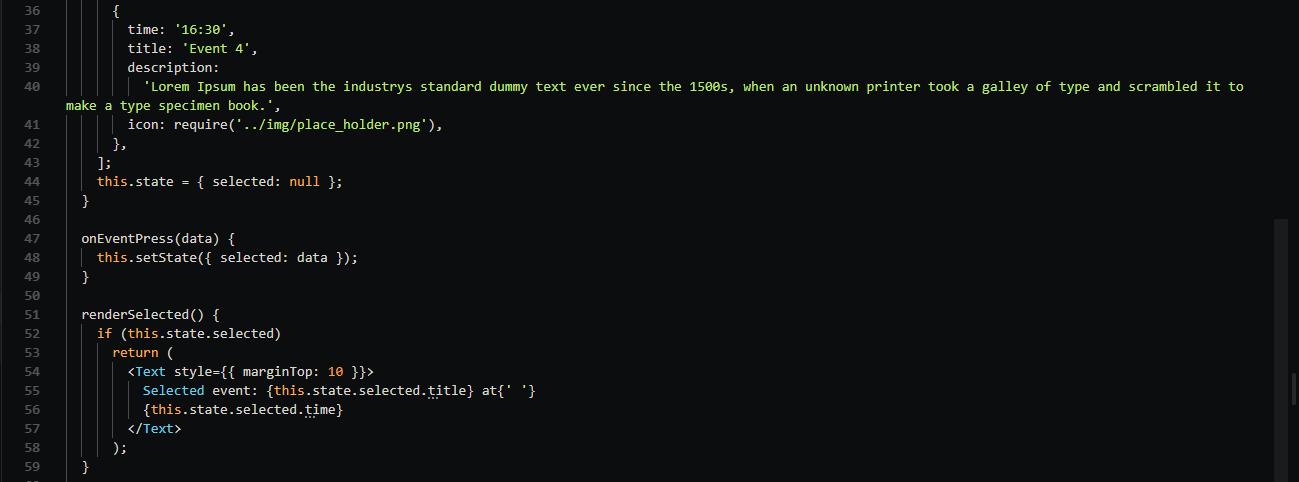
******

******

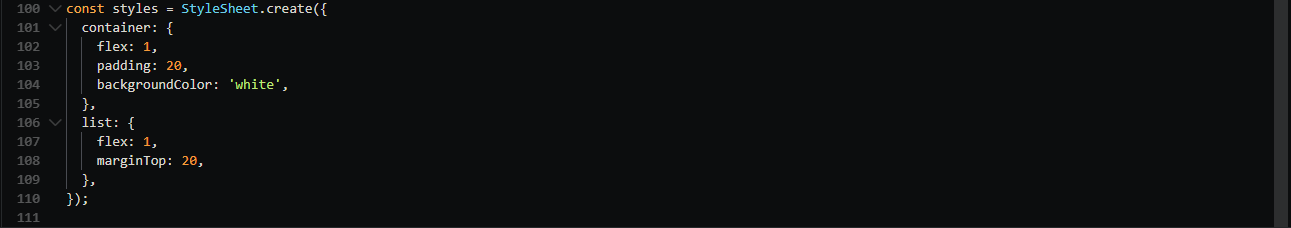
******

***TimeLinePress:***

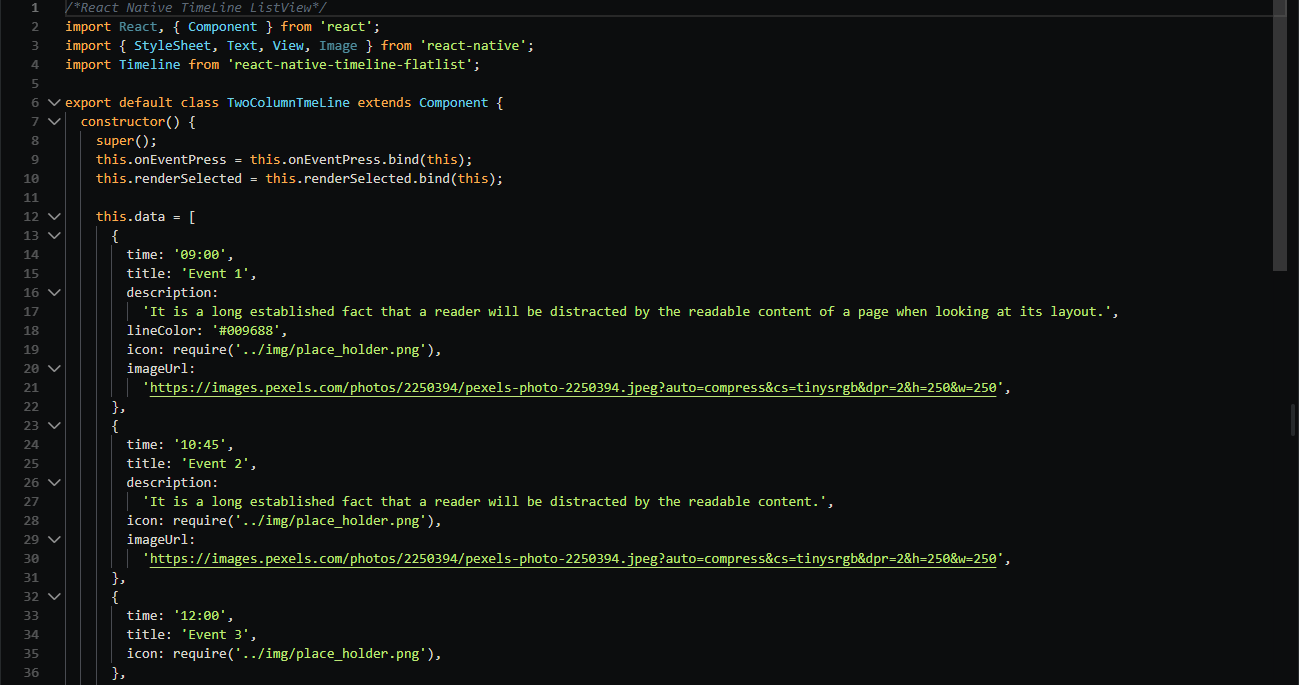
******

******

******

******

***TwoColumnTimeLine:***

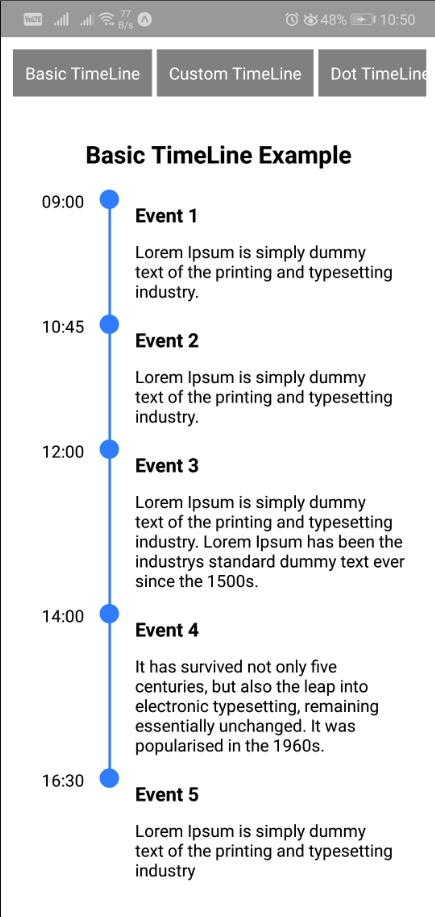
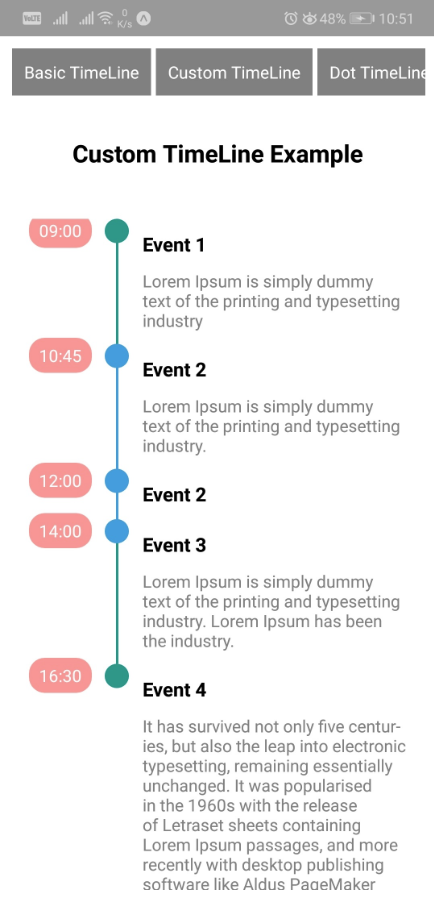
******

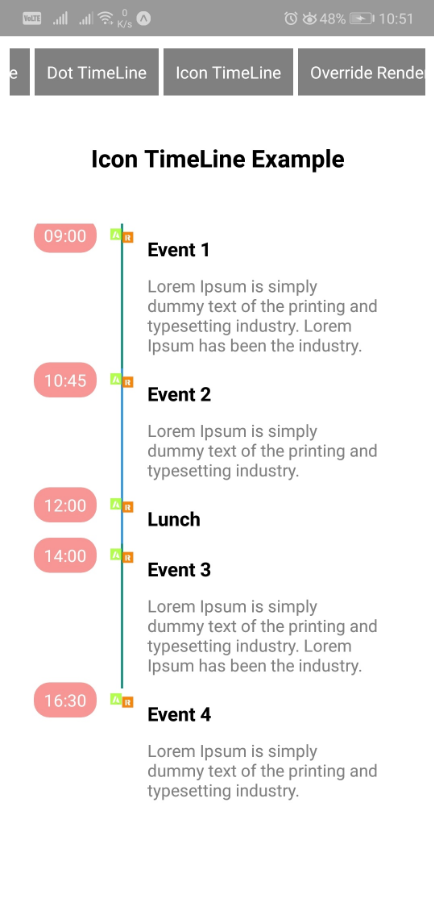
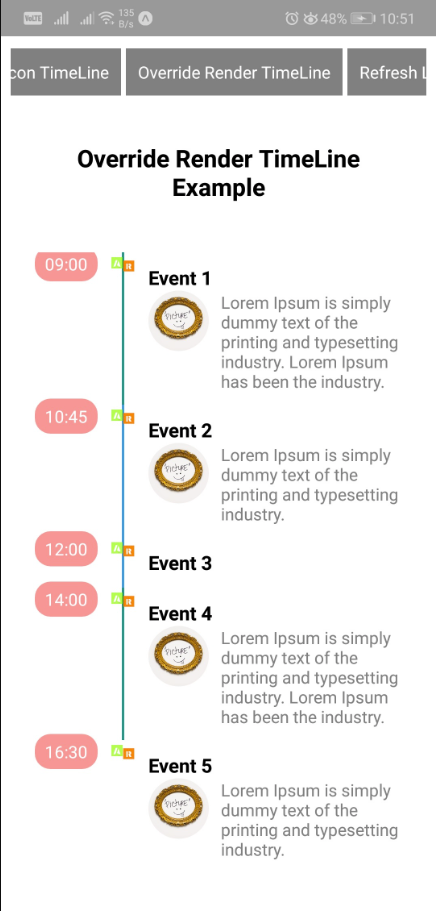
******

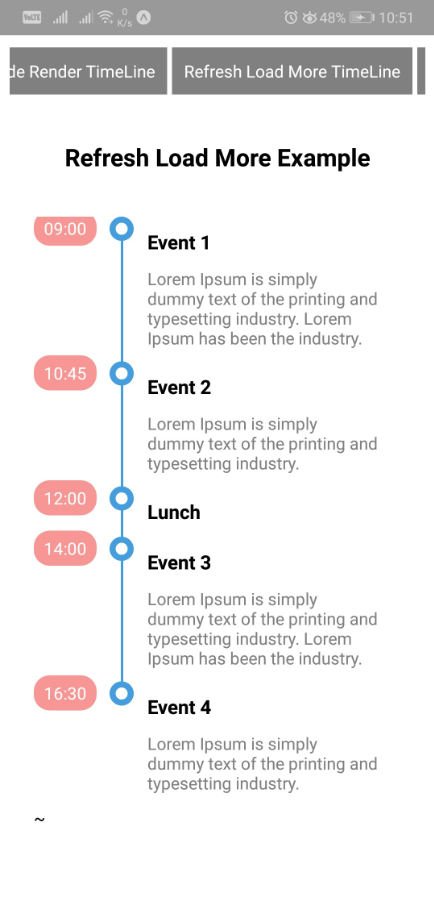
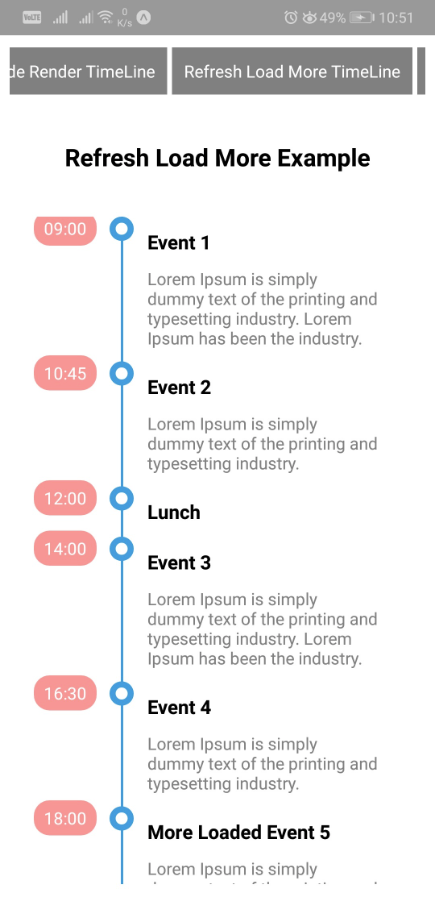
******

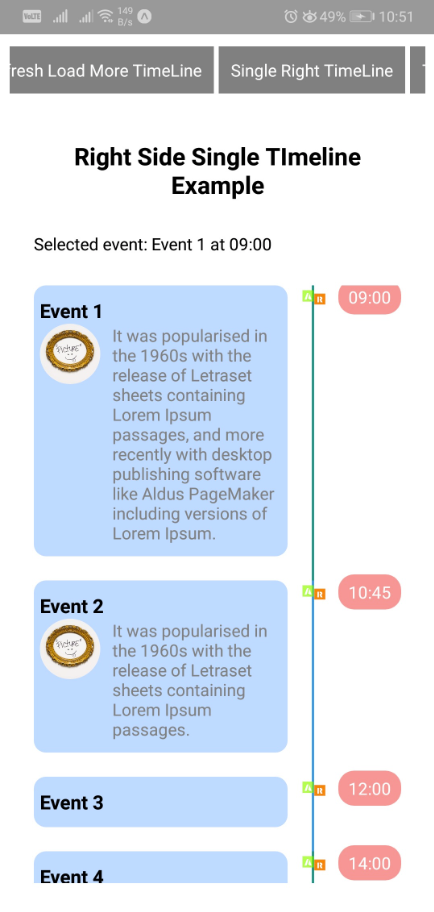
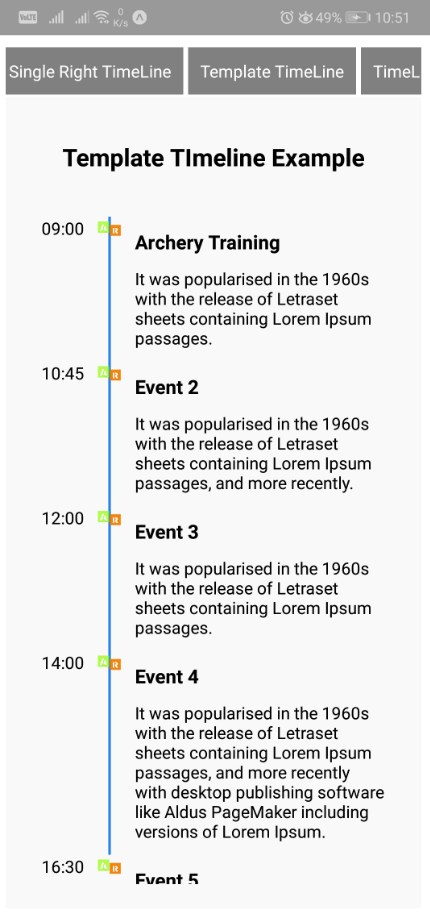
******

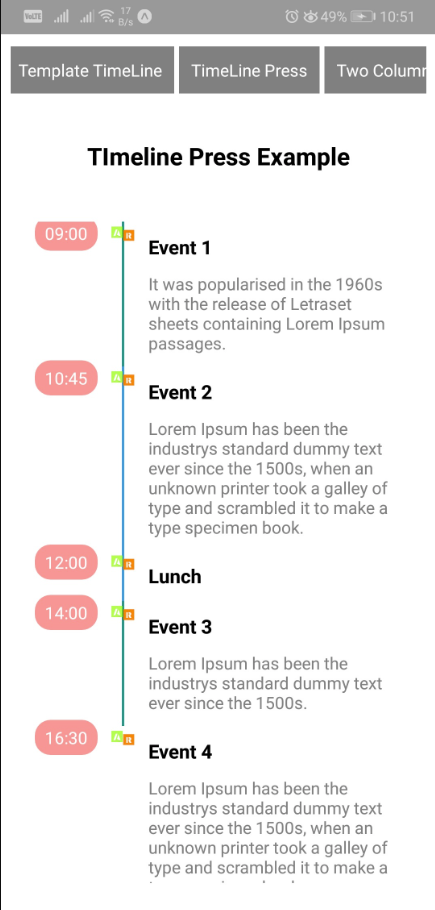
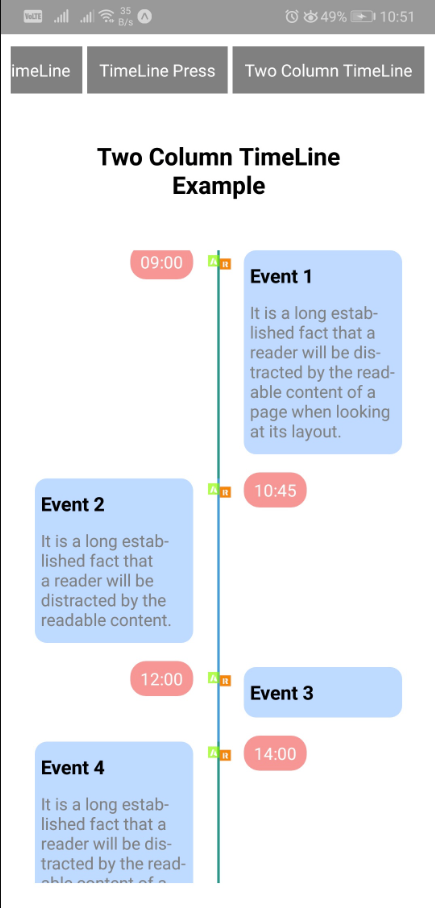
### Hình ảnh kết quả:

**** ****

**** ****

**** ****

**** ****

**** ****

# GridView Components

## GridView:

GridView cho phép hiển thị dữ liệu trong dạng lưới (grid). GridView rất hữu ích cho việc trình bày hình ảnh, sản phẩm, hoặc bất kỳ thông tin nào cần được sắp xếp theo hàng và cột. Thành phần này giúp người dùng dễ dàng duyệt qua một lượng lớn thông tin một cách trực quan và hấp dẫn.

GridView có cùng mức độ phổ biến như Listview, cả hai đều được sử dụng để sắp xếp các mục theo cách phù hợp. GridView có thể được sử dụng khi chúng ta phải tạo một View Group hiển thị các mục trong lưới hai chiều có thể cuộn. Ví dụ hữu ích nhất của Grid là thư viện ảnh, nơi chúng ta phải giới thiệu tất cả các hình ảnh.

### Cấu trúc:

GridView thường sử dụng FlatList hoặc ScrollView kết hợp với các kiểu hiển thị khác nhau để sắp xếp các mục trong một lưới. Dưới đây là một ví dụ đơn giản về cách xây dựng một GridView trong React Native:

### Cách hiện thực hóa:

Import thư viện:

*import { FlatList} from 'react-native'*

Rendering element:

*<FlatList*

*data={this.state.dataSource}*

*renderItem={({ item }) => (*

*<View*

*style={{*

*flex: 1,*

*flexDirection: 'column',*

*margin: 1*

*}}>*

*<Image*

*style={styles.imageThumbnail}*

*source={{ uri: item.src }}*

*/>*

*</View>*

*)}*

*//Setting the number of column*

*numColumns={3}*

*keyExtractor={(item, index) => index.toString()}*

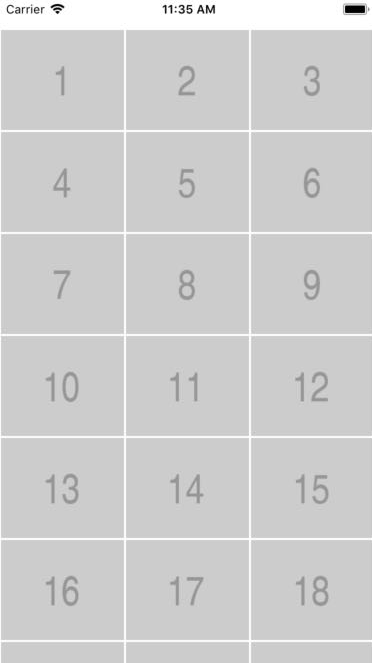
*/>*

### Mã nguồn ví dụ:

****

****

### Hình ảnh kết quả:

****

## View Like Google Play Store:

Giao diện của Google Play Store là một ví dụ điển hình về cách trình bày thông tin ứng dụng một cách trực quan và hấp dẫn. Mô hình này thường bao gồm một danh sách các ứng dụng với hình ảnh, tên ứng dụng, đánh giá và mô tả ngắn được thiết kế theo hướng Chế độ xem cuộn dọc và ngang

### Cấu trúc:

Giao diện tương tự Google Play Store có thể được xây dựng bằng cách sử dụng FlatList hoặc ScrollView để hiển thị các ứng dụng

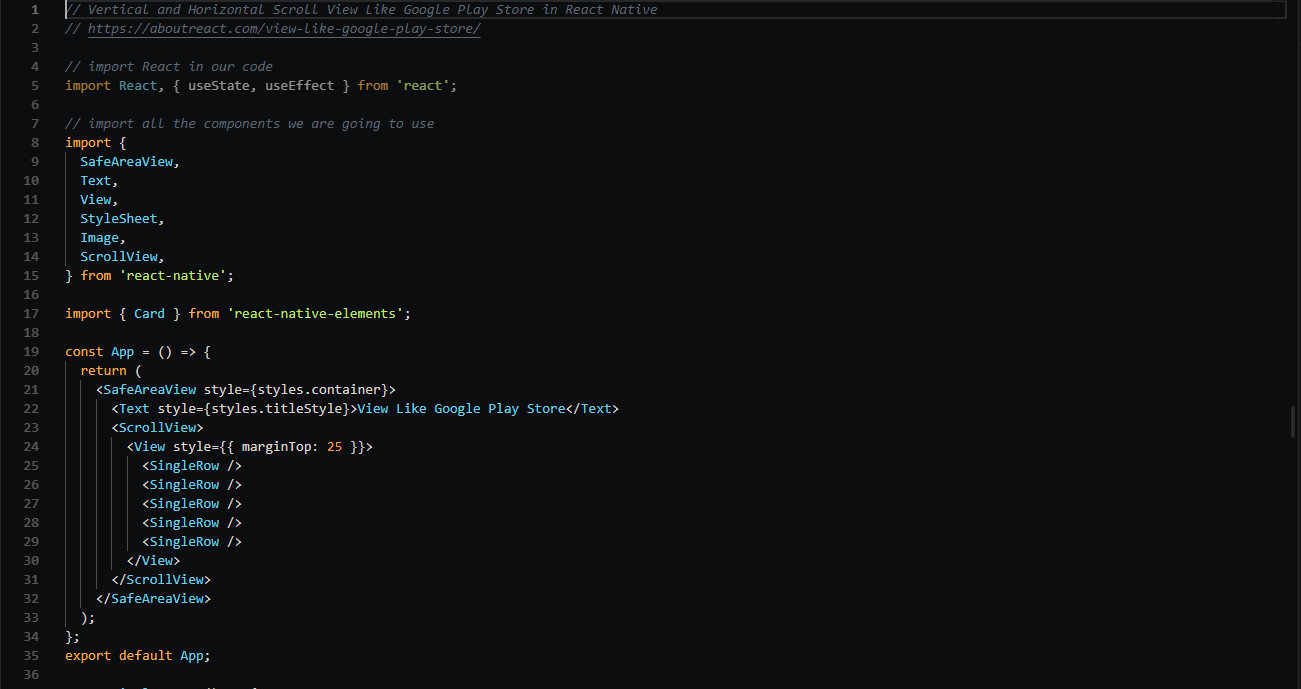
### Cách hiện thực hóa:

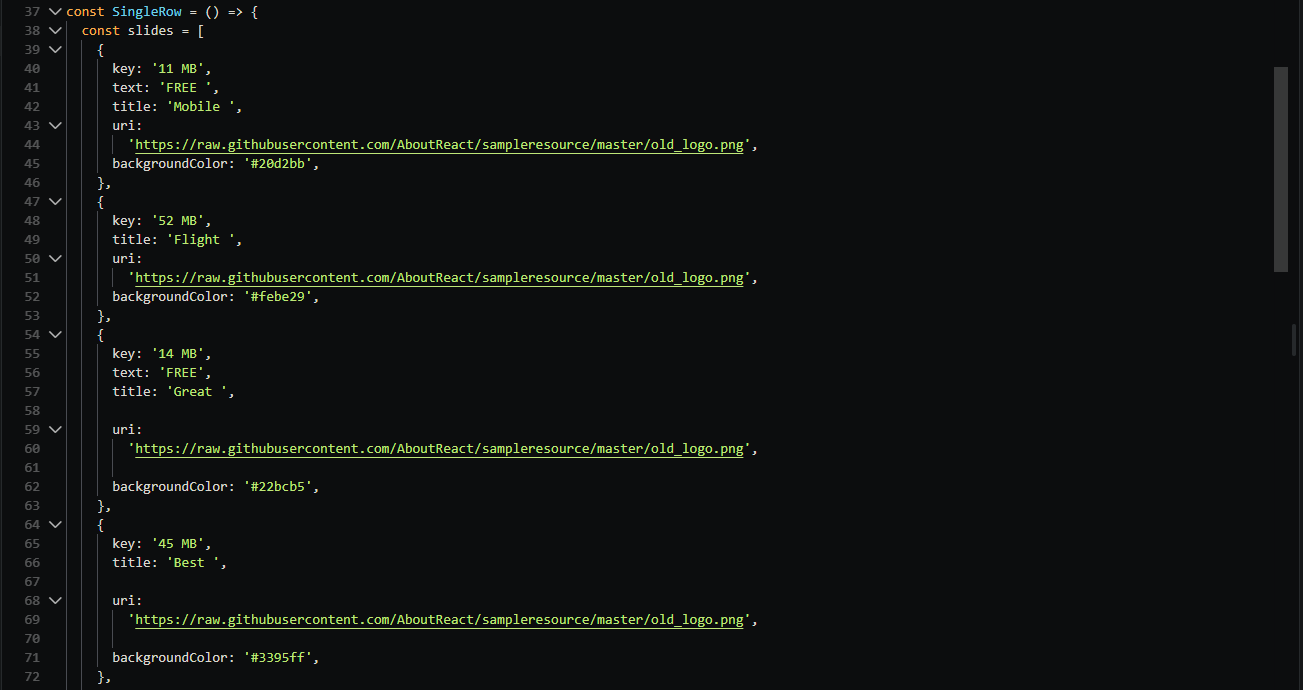
Import thư viện:

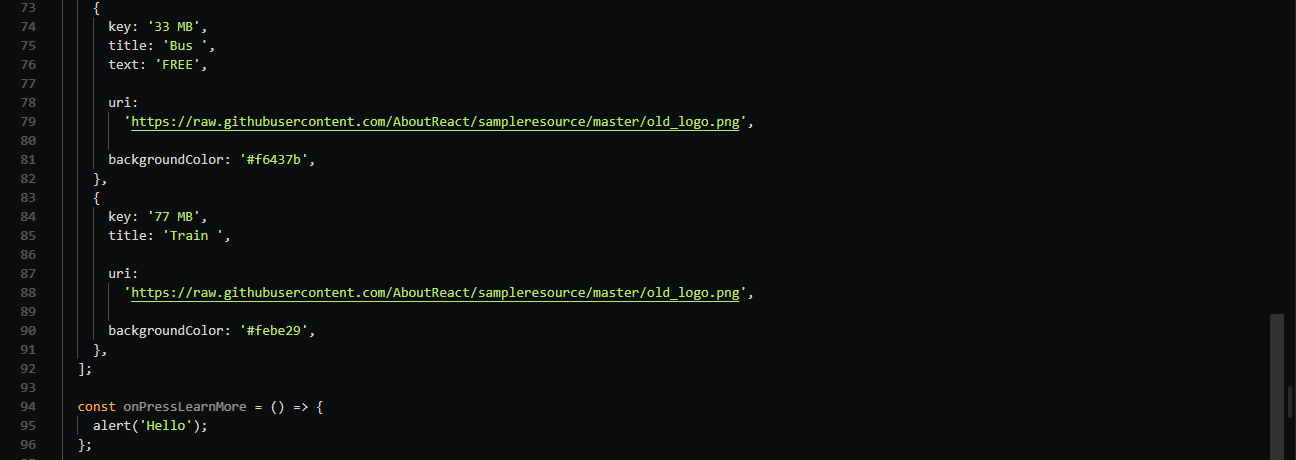
*npm install @rneui/themed –save*

*npx pod-install*

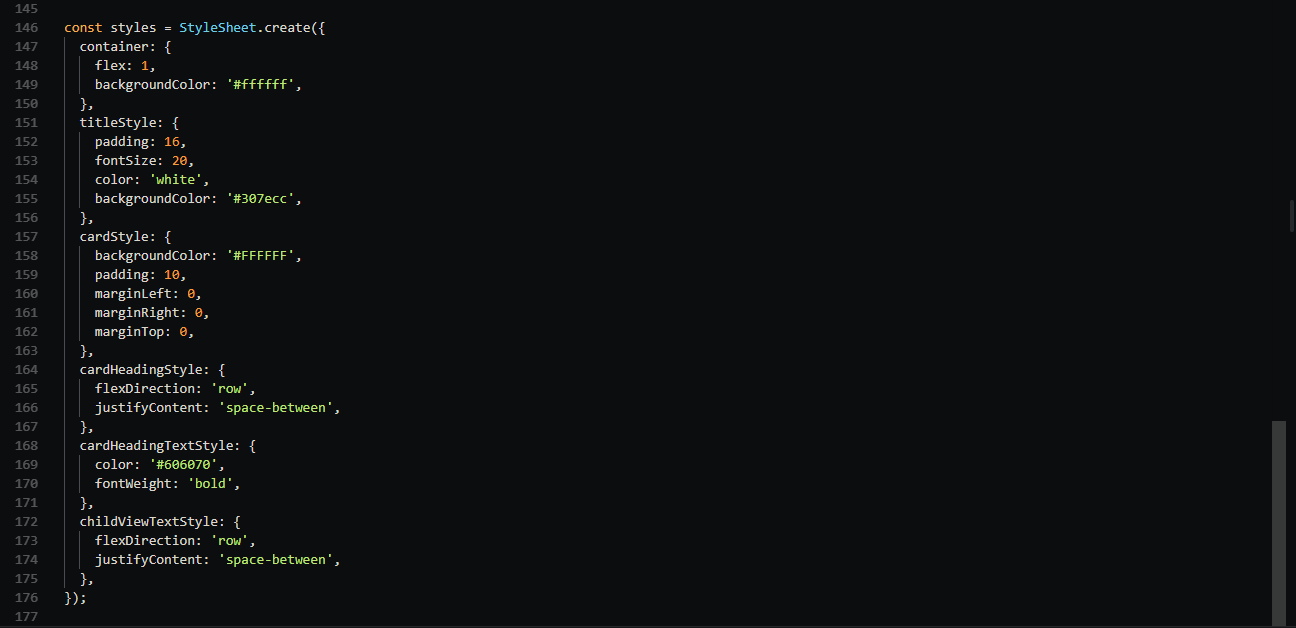
### Mã nguồn ví dụ:

**

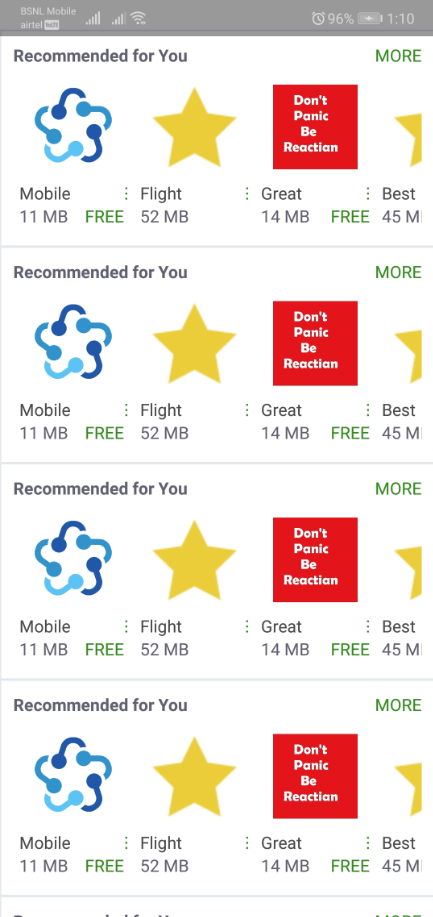
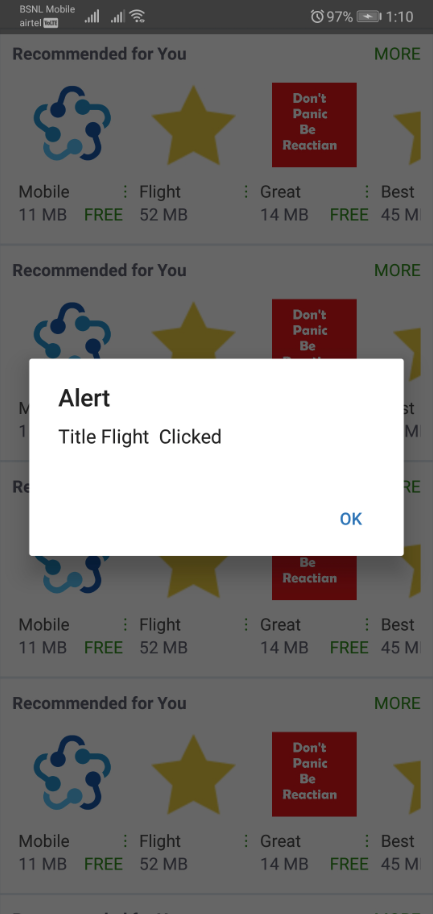
**

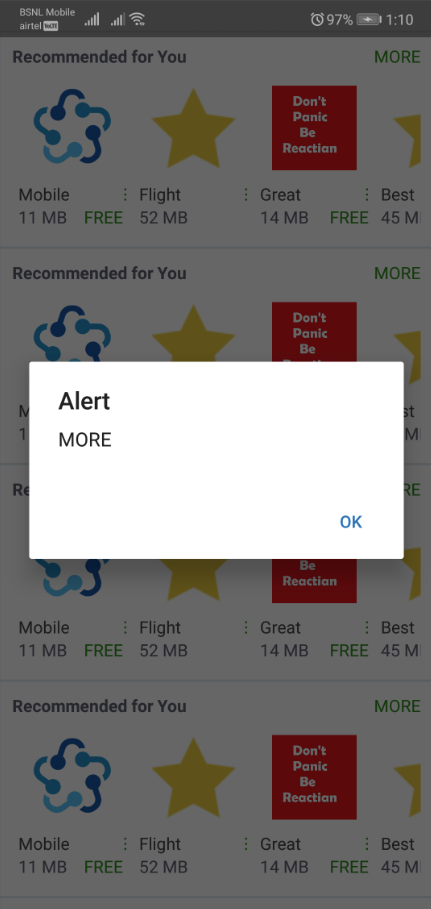
** **

**

**

### Hình ảnh kết quả:

** **

**

# Expandable - Collapsible Components

## Tree View:

Tree View là một cấu trúc dữ liệu hiển thị thông tin theo dạng phân cấp, cho phép người dùng dễ dàng duyệt qua các mục cha và con. Trong React Native, việc triển khai một Tree View có thể rất hữu ích cho các ứng dụng cần tổ chức thông tin phức tạp, chẳng hạn như quản lý tệp tin, danh sách sản phẩm, hoặc thậm chí là cây cấu trúc dữ liệu.

### Cấu trúc và Tính năng:

Một Tree View thường bao gồm các tính năng như:

Mở rộng/Thu gọn: Người dùng có thể mở rộng hoặc thu gọn các mục cha để xem hoặc ẩn các mục con.

Tương tác: Người dùng có thể nhấn vào các mục để thực hiện các hành động khác nhau.

Tùy biến: Cấu trúc có thể được tùy chỉnh theo nhu cầu của ứng dụng.

### Cách hiện thực hóa:

Import thư viện:

*import TreeView from 'react-native-final-tree-view';*

Rendering element:

*<TreeView*

*data={state.data}*

*renderNode={({ node, level, isExpanded, hasChildrenNodes }) => {*

*return (*

*<View>*

*<Text*

*style={{*

*marginLeft: 25 \* level,*

*fontSize: 18,*

*}}>*

*{getIndicator(isExpanded, hasChildrenNodes)} {node.name}*

*</Text>*

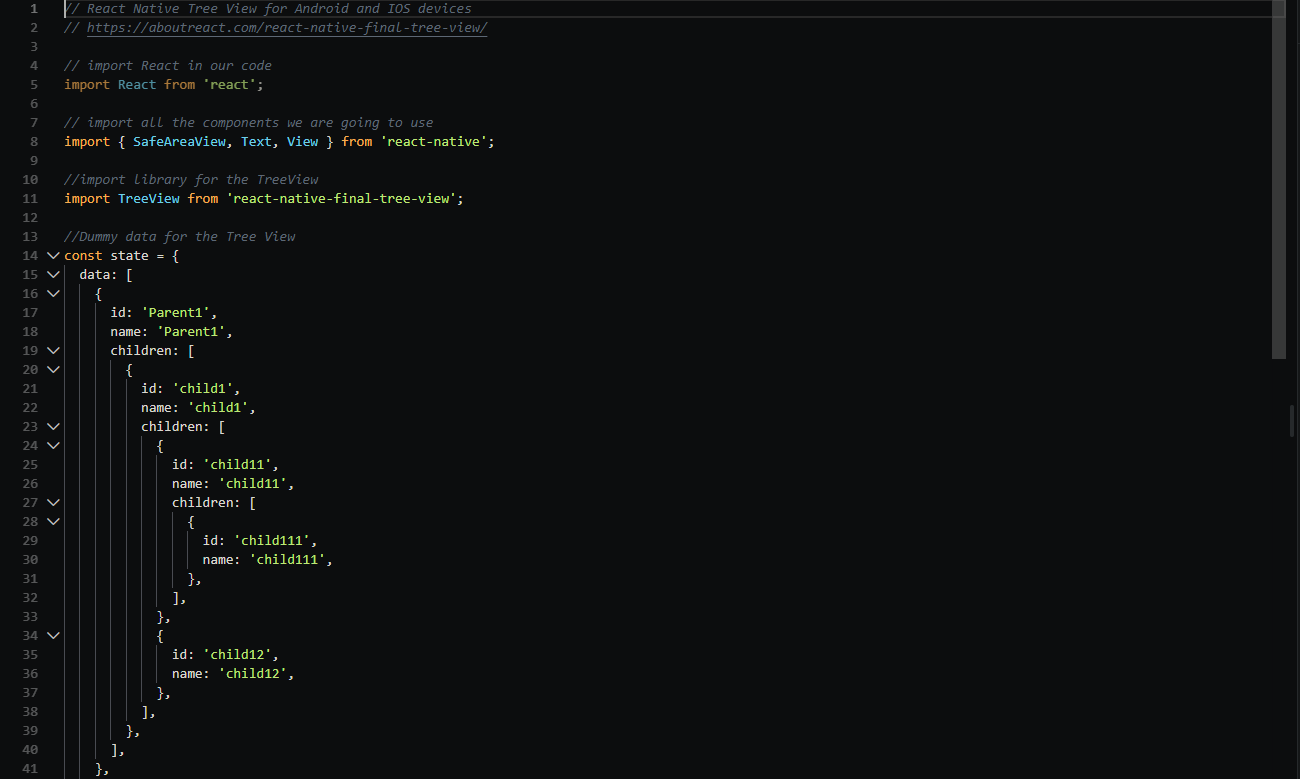
*</View>*

*);*

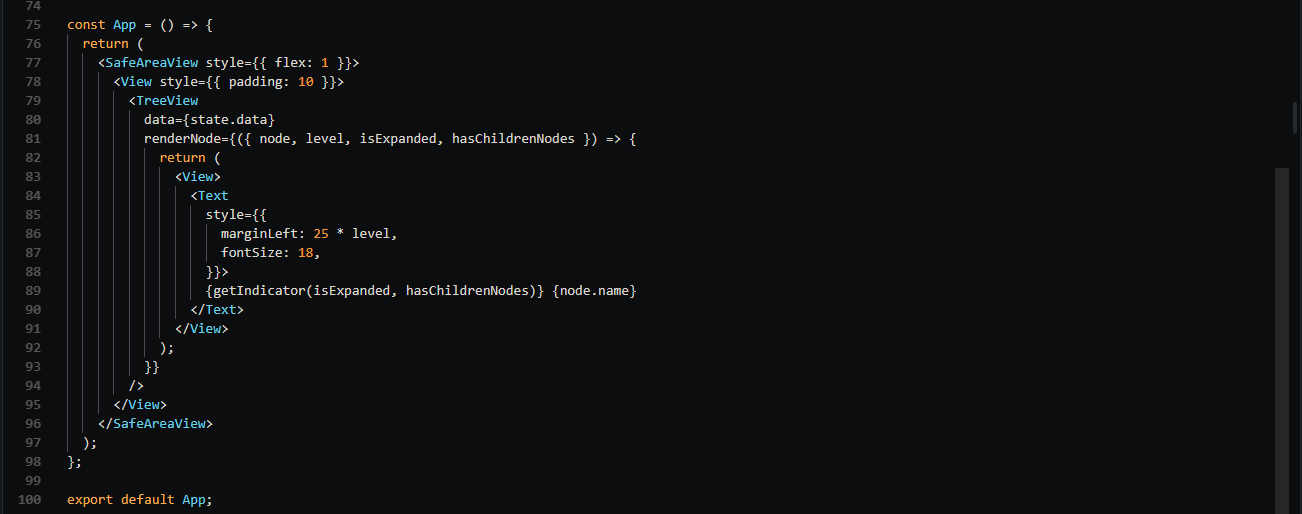
*}}*

*/>*

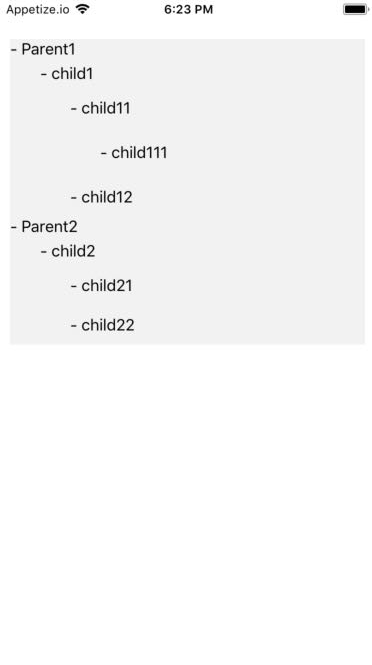
### Mã nguồn ví dụ:







### Hình ảnh kết quả:

## Collapsible/Accordion:

Collapsible hay Accordion là một thành phần giao diện người dùng phổ biến, cho phép người dùng mở hoặc đóng các mục để xem nội dung bên trong. Điều này giúp tiết kiệm không gian và tạo ra một trải nghiệm tương tác thân thiện cho người dùng.

### Cấu trúc và Tính năng:

Một Accordion thường bao gồm các tính năng như:

Mở rộng/Thu gọn: Người dùng có thể nhấn vào tiêu đề để mở hoặc đóng nội dung.

Chỉ một mục mở: Trong một số trường hợp, chỉ cho phép một mục mở cùng lúc.

Tùy chỉnh: Có thể tùy chỉnh nội dung bên trong và tiêu đề theo nhu cầu.

### Cách hiện thực hóa:

Import thư viện:

*npm install react-native-collapsible –save*

*npm install react-native-collapsible/Accordion –save*

*npm install react-native-animatable --save*

Rendering element:

**Collapsible**

*{/\*Code for Single Collapsible Start\*/}*

*<TouchableOpacity onPress={this.toggleExpanded}>*

*<View style={styles.header}>*

*<Text style={styles.headerText}>Single Collapsible</Text>*

*{/\*Heading of Single Collapsible\*/}*

*</View>*

*</TouchableOpacity>*

*{/\*Content of Single Collapsible\*/}*

*<Collapsible collapsed={this.state.collapsed} align="center">*

*<View style={styles.content}>*

*<Text style={{ textAlign: 'center' }}>*

*This is a dummy text of Single Collapsible View*

*</Text>*

*</View>*

*</Collapsible>*

*{/\*Code for Single Collapsible Ends\*/}*

**Accordion**

*<Accordion*

*activeSections={activeSections}*

*// For any default active section*

*sections={CONTENT}*

*// Title and content of accordion*

*touchableComponent={TouchableOpacity}*

*// Which type of touchable component you want*

*// It can be the following Touchables*

*// TouchableHighlight, TouchableNativeFeedback*

*// TouchableOpacity , TouchableWithoutFeedback*

*expandMultiple={multipleSelect}*

*// Do you want to expand mutiple at a time or single at a time*

*renderHeader={this.renderHeader}*

*// Header Component(View) to render*

*renderContent={this.renderContent}*

*// Content Component(View) to render*

*duration={400}*

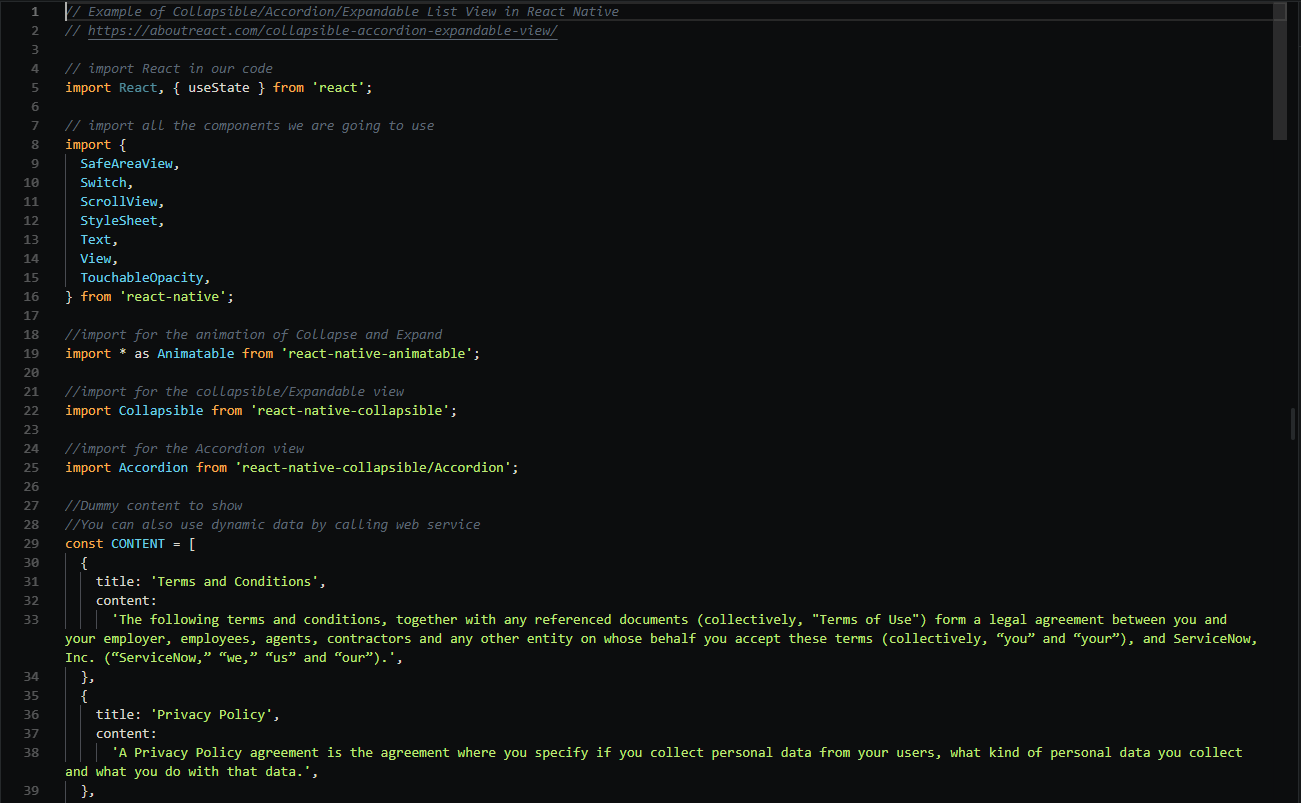
*// Duration for Collapse and expand*

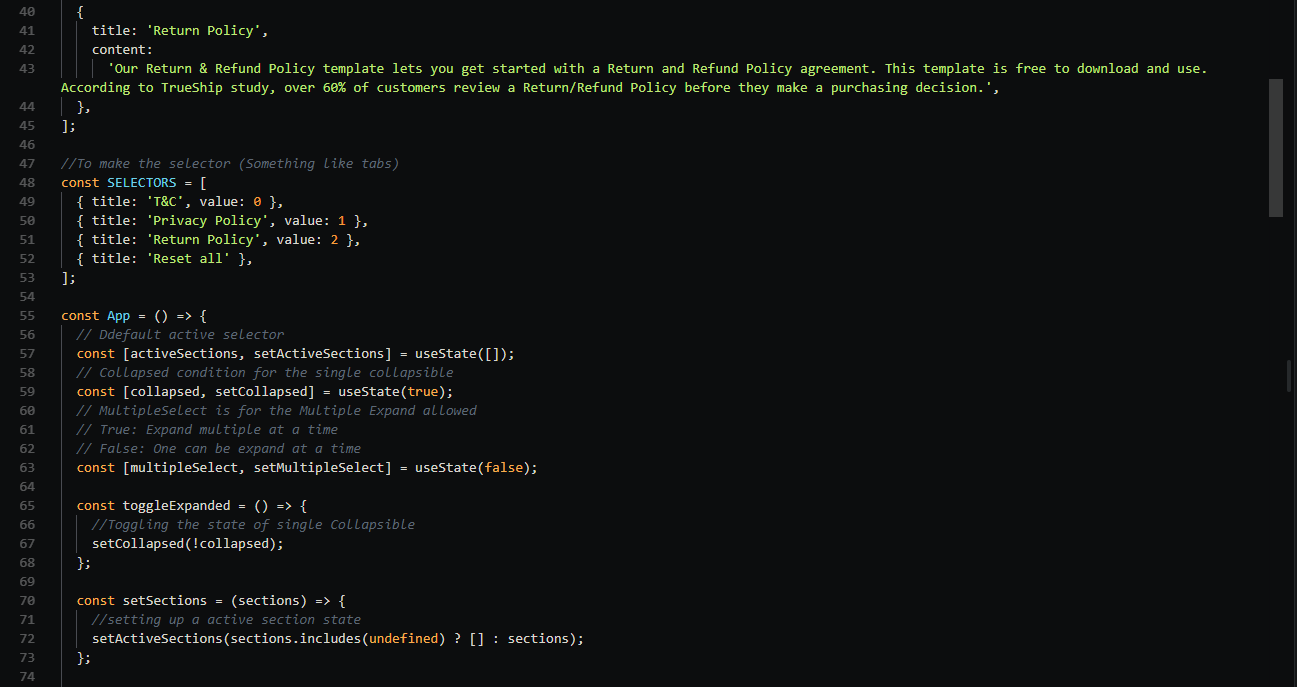
*onChange={setSections}*

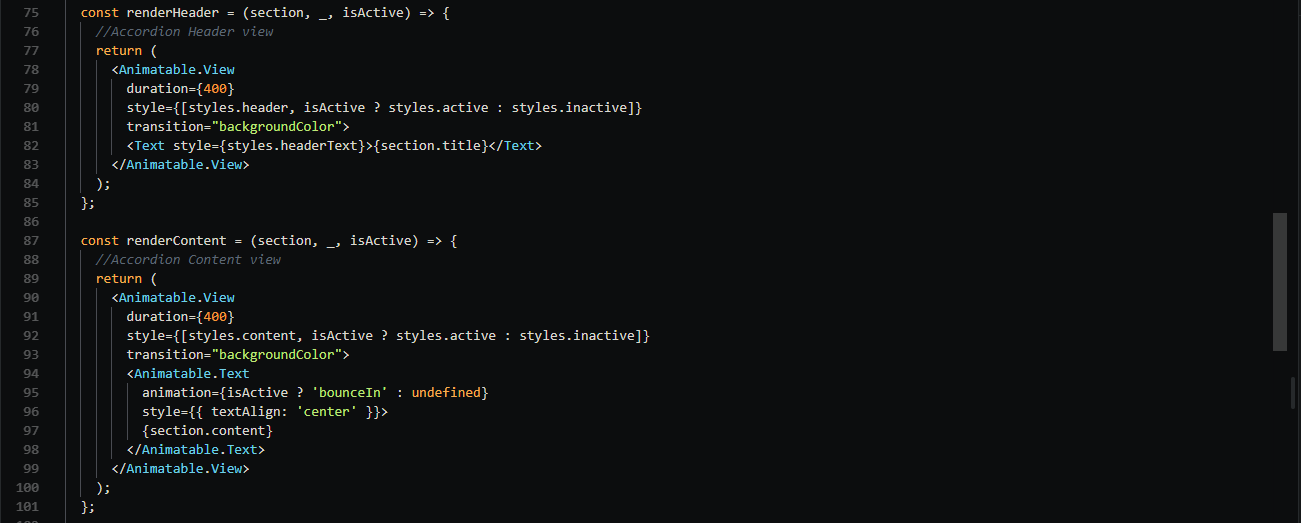
*// Setting the state of active sections*

*/>*

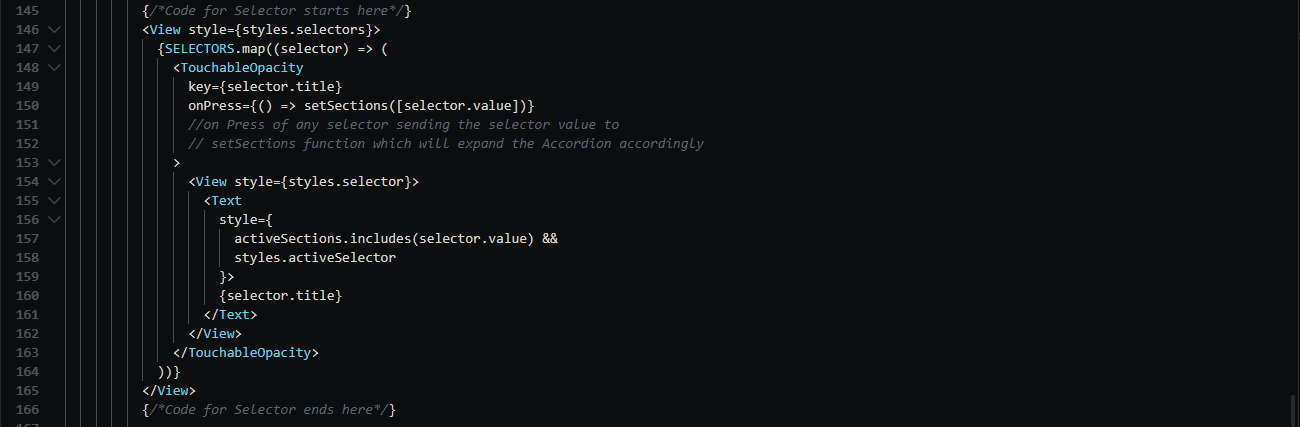
### Mã nguồn ví dụ:

****

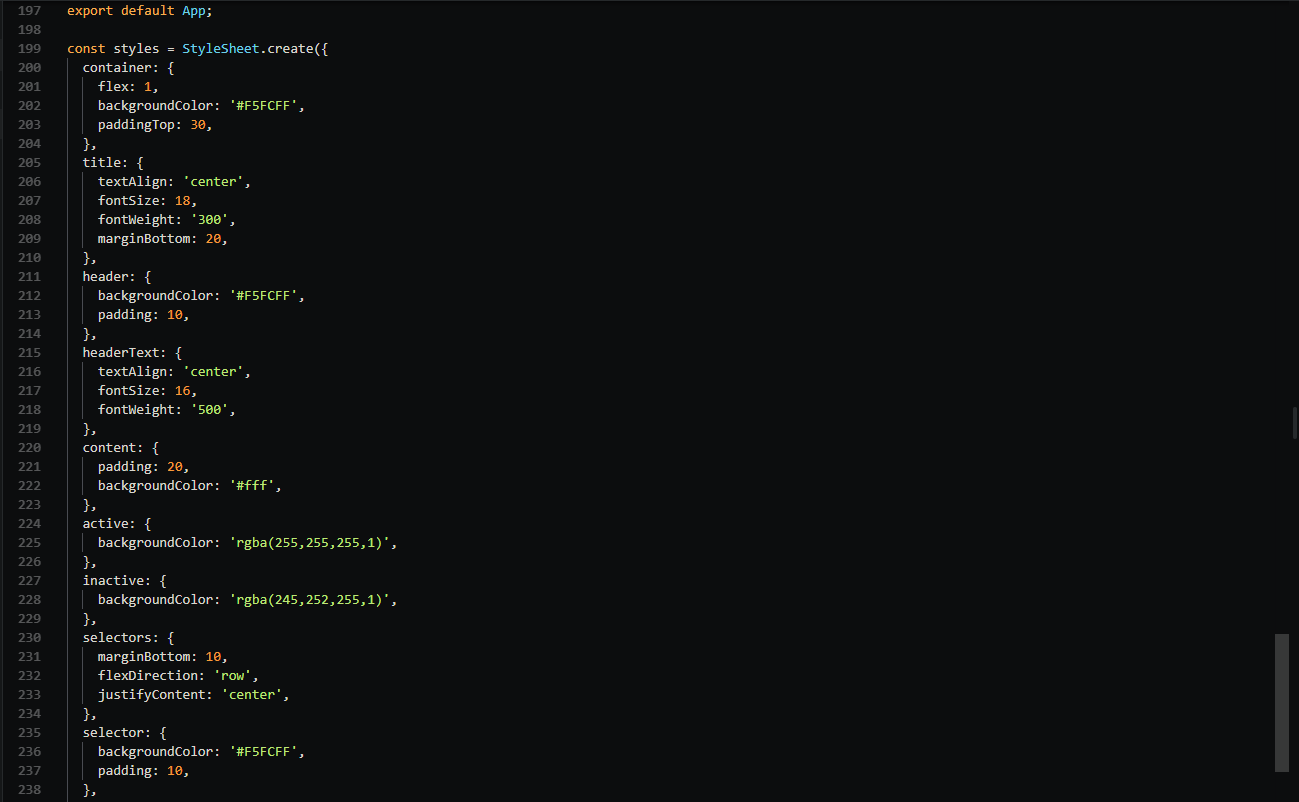
****

****

****

****

****

****

****

### Hình Ảnh kết quả:

