



The Travel Companion App! This app helps you visualize your travel journey from one destination to another, tracking your stops along the way. Whether you're going from Delhi to Bangalore or any other destinations, this app has got you covered.

Introduction

The Travel Companion App is designed to provide a smooth and interactive experience for users to track their travel progress. It offers two main views: Lazy List and Normal List, allowing users to choose their preferred way of viewing their travel stops. Additionally, users can toggle between displaying distances in kilometers or miles.

Features

Lazy List: Provides a lazy-loaded list of travel stops. Ideal for large datasets, ensuring smooth scrolling performance.

Normal List: Offers a standard list of travel stops. Suitable for smaller datasets or users who prefer traditional list views.

Distance Conversion: The app seamlessly converts distances between kilometers and miles, giving users flexibility in their preferred unit of measurement.

Interactive Progress Tracking: Users can interactively track their travel progress by checking off visited stops, with the app automatically updating the total and remaining distances accordingly.

Usage

To use the Travel Companion App, simply follow these steps:

1. Launch the App: Upon launching the app, you'll be greeted with the main interface displaying your travel journey from Delhi to Bangalore.

2. Viewing Options: Choose between Lazy List and Normal List by tapping on the corresponding section headers.
3. Distance Units: Toggle between kilometers and miles by activating the "Convert to Miles" switch.
4. Tracking Progress: Check off visited stops to track your progress visually. The app dynamically updates the total and remaining distances based on your selections.

Code Overview

MainActivity

The MainActivity initializes the app by generating random distances for travel stops and setting the content to the TravelPreview composable function.

Composable Functions

ScrollableRow: Renders a lazy-loaded list of travel stops with a linear progress indicator, ideal for smooth scrolling performance.

NormalRow: Displays a standard list of travel stops with a linear progress indicator, suitable for smaller datasets or traditional list views.

RowItem: Represents a single travel stop item within a list, allowing users to check off visited stops.

UnitRow: Renders a row with a switch to toggle between kilometers and miles.

Travel: Builds the main UI layout, allowing users to switch between Lazy List and Normal List views and toggle distance units.

TravelPreview: Entry point for the app, setting up the main Travel composable function with random distances.

Utility Functions

generateRandomDoubles: Generates a list of random distances for travel stops.

convertDistanceToMiles: Converts distances from kilometers to miles.

Conclusion

The Travel Companion App offers a user-friendly interface for tracking travel progress, with options for lazy-loaded or standard list views and flexible distance unit conversion. Whether you're embarking on a long journey or a short trip, this app is your perfect travel companion. Enjoy your travels!

Screenshots on the Next Page:

```
Image(painter = painterResource(id = R.drawable.direction), contentDescription = null, modifier = Modifier.padding(8.dp))
}
NormalRow(items = travelStops, showDistanceInMiles = miles)
UnitRow(roundUp = miles, onRoundUpChanged = {miles=it}, modifier=modifier)
```

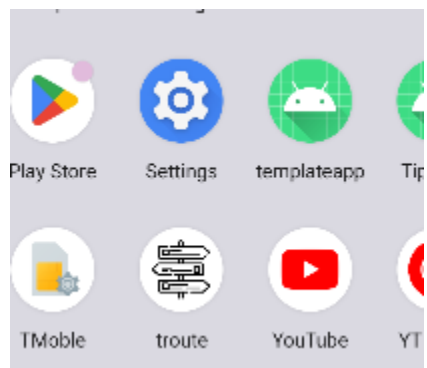
```
Row(
    modifier = Modifier.horizontalScroll(rememberScrollState())
) { this: RowScope
    for (index in checkedStateList.indices) {
        RowItem(
            travelStop = items[index],
            checked = checkedStateList[index],
            onCheckedChange = { isChecked -> onCheckedChange(index, isChecked) },
            modifier = Modifier.padding(8.dp)
        )
    }
}
```

```
Image(painter = painterResource(id = R.drawable.direction), contentDescription = null, modifier = Modifier.padding(8.dp))
}
ScrollableRow(items = travelStops, showDistanceInMiles = miles)
```

```

        .height(4.dp)
    )
    LazyRow { this: LazyListScope
        items(checkedStateList.size) { this: LazyItemScope index ->
            RowItem(
                travelStop = items[index],
                checked = checkedStateList[index],
                onCheckedChange = { isChecked -> onCheckedChange(index, isChecked) },
                modifier = Modifier.padding(8.dp)
            )
        }
    }
}

```



6:07



Delhi to Bangalore

LazyList 



TravelStop1

Distance: 43.66



TravelStop2

Distance: 29.21

Total Distance: 597.87 Km

Traveled Distance: 0.00 Km

Remaining Distance: 597.87 Km

bloody Travelled Distance: 0.00 Km

NormalList 



TravelStop1

Distance: 43.66



TravelStop2

Distance: 29.21

Total Distance: 597.87 Km

Traveled Distance: 0.00 Km

Remaining Distance: 597.87 Km

bloody Travelled Distance: 0.00 Km

Convert to Miles



6:08

Deini to Bangalore

LazyList 



TravelStop1

Distance: 43.66



TravelStop2

Distance: 29.21

Total Distance: 597.87 Km

Traveled Distance: 43.66 Km

Remaining Distance: 554.21 Km

bloody Travelled Distance: 43.66 Km

NormalList 



TravelStop1

Distance: 43.66



TravelStop2

Distance: 29.21

Total Distance: 597.87 Km

Traveled Distance: 72.94 Km

Remaining Distance: 524.93 Km

bloody Travelled Distance: 72.94 Km

Convert to Miles



6:08

Deini to Bangalore

LazyList



TravelStop11

Distance: 8.18



TravelStop12

Distance: 44.01

Total Distance: 371.50 Miles
Traveled Distance: 371.50 Miles
Remaining Distance: 0.00 Miles
bloody Travelled Distance: 597.87
Km

NormalList



TravelStop11

Distance: 8.18



TravelStop12

Distance: 44.01

Total Distance: 371.50 Miles
Traveled Distance: 327.49 Miles
Remaining Distance: 44.01 Miles
bloody Travelled Distance: 527.05
Km

Convert to Miles

