Application-Distract into direction

**An AI (voice)which guides humans towards their goals**

**Guiding him a direction towards his passion**

Shifting your energy

Do you have a passion/What is your passion? Must include Psychology

What is your goal?

What do you want to accomplish in life

* How passionate are you on metric, (let’s say people want to do it in 6 months little patient, more passionate, 1 month too eager\_) on achieving your goal

How do you stay on track

* Accountability partner
* Deadlines
* Losing time each day- since you must accomplish this in a certain time|

How do you measure skill level?

1-10

How much time will dedicate for your task every day?

Time

Did you invest your time?

Many tasks each day to keep you focused on productivity!

Prediction of your future time-day/week based on your present time.

**Good Code:**

import React, { useState } from 'react';

import { View, Text, TextInput, Button, TouchableOpacity, StyleSheet } from 'react-native';

export default function App() {

const [step, setStep] = useState(1);

const [passion, setPassion] = useState('');

const [noPassion, setNoPassion] = useState(false);

const [selectedPassion, setSelectedPassion] = useState('');

const [commitment, setCommitment] = useState(null);

const [learningLevel, setLearningLevel] = useState('');

const passionOptions = ['Coding', 'Photography', 'Writing', 'Fitness', 'Music'];

const commitmentOptions = [1, 2, 3, 4, 5, 6, 7, 8];

const learningLevels = ['Basic', 'Intermediate', 'Advanced'];

return (

<View style={styles.container}>

{*/\* Question 1: Ask about passion \*/*}

{step === 1 && (

<>

<Text style={styles.question}>What is your passion?</Text>

<TextInput

style={styles.input}

placeholder="Type your passion here"

value={passion}

onChangeText={setPassion}

/>

<Button title="Next" onPress={() => setStep(2)} disabled={!passion} />

<Text style={styles.orText}>OR</Text>

<Button title="I don't have one" onPress={() => { setNoPassion(true); setStep(2); }} />

</>

)}

{*/\* Question 2: Show options if they don't have a passion \*/*}

{step === 2 && noPassion && (

<>

<Text style={styles.question}>Select a passion from below:</Text>

{passionOptions.map((p) => (

<TouchableOpacity key={p} onPress={() => { setSelectedPassion(p); setStep(3); }} style={styles.optionButton}>

<Text style={styles.optionText}>{p}</Text>

</TouchableOpacity>

))}

</>

)}

{step === 2 && !noPassion && (

<>

<Text style={styles.responseText}>Your passion: {passion}</Text>

<Button title="Next" onPress={() => setStep(3)} />

</>

)}

{*/\* Question 3: Daily Commitment \*/*}

{step === 3 && (

<>

<Text style={styles.question}>How many hours per day can you commit?</Text>

{commitmentOptions.map((hour) => (

<TouchableOpacity key={hour} onPress={() => { setCommitment(hour); setStep(4); }} style={styles.optionButton}>

<Text style={styles.optionText}>{hour} hrs</Text>

</TouchableOpacity>

))}

</>

)}

{*/\* Question 4: Learning Level \*/*}

{step === 4 && (

<>

<Text style={styles.question}>Select your learning level</Text>

{learningLevels.map((level) => (

<TouchableOpacity key={level} onPress={() => { setLearningLevel(level); setStep(5); }} style={styles.optionButton}>

<Text style={styles.optionText}>{level}</Text>

</TouchableOpacity>

))}

</>

)}

{*/\* Final Step: Show Summary \*/*}

{step === 5 && (

<>

<Text style={styles.responseText}>Summary:</Text>

<Text style={styles.summaryText}>Passion: {selectedPassion || passion}</Text>

<Text style={styles.summaryText}>Commitment: {commitment} hrs/day</Text>

<Text style={styles.summaryText}>Learning Level: {learningLevel}</Text>

<Button title="Restart" onPress={() => setStep(1)} />

</>

)}

</View>

);

}

*// Styles for the app*

const styles = StyleSheet.create({

container: {

flex: 1,

justifyContent: 'center',

alignItems: 'center',

padding: 20,

backgroundColor: '#f5f5f5',

},

question: {

fontSize: 20,

fontWeight: 'bold',

marginBottom: 20,

textAlign: 'center',

},

input: {

borderWidth: 1,

borderColor: '#ccc',

padding: 10,

width: '80%',

borderRadius: 5,

backgroundColor: 'white',

marginBottom: 10,

},

orText: {

fontSize: 16,

marginVertical: 10,

},

optionButton: {

padding: 15,

backgroundColor: '#007BFF',

borderRadius: 5,

marginVertical: 5,

width: '80%',

alignItems: 'center',

},

optionText: {

fontSize: 16,

color: 'white',

},

responseText: {

fontSize: 18,

fontWeight: 'bold',

marginTop: 20,

textAlign: 'center',

},

summaryText: {

fontSize: 16,

marginVertical: 5,

},

});

1. **Personalized Resource Recommendations:**
   * Based on the user's passion and learning level, provide relevant resources (e.g., websites, tutorials, books).
   * This is the logical next step, the app is gathering data, but not using it to provide value.
2. **Daily/Weekly Goals:**
   * Allow users to set daily or weekly goals related to their passion.
   * Implement a progress tracking system.
3. **Reminders and Notifications:**
   * Send reminders to users to work on their passion.
   * Local notifications can be implemented.
4. **Community Features:**
   * A forum or chat where users can connect with others who share their passion.
   * This could be a large undertaking.
5. **Progress Tracking and Analytics:**
   * Visualize the user's progress over time.
   * Graphs and charts.
6. **Integration with External Services:**
   * Connect with APIs for learning platforms (e.g., YouTube, Udemy).
   * This would allow for content recommendations.
7. **Customizable Passion List:**
   * Allow the user to add passions that are not on the provided list.
8. **Calendar View:**
   * Display the user's commitment hours and goals in a calendar format.
9. **User Profiles:**
   * Allow users to create profiles with more detailed information.
10. **Refined Passion Selection:**
    * Instead of a simple list, implement a search or category-based passion selection.
11. **Onboarding Improvements:**
    * Add more detailed descriptions of what each question is asking.
    * Add visual aids to the onboarding process.

**Prioritization:**

* **Immediate:**
  + Personalized Resource Recommendations (this provides immediate value).
  + Daily/Weekly Goals (helps users stay on track).
* **Medium:**
  + Reminders and Notifications.
  + Progress Tracking and Analytics.
  + Customizable Passion List.
* **Long-term:**
  + Community Features.
  + Integration with External Services.
  + User Profiles.

Try this too

import React, { useState } from 'react';

import {

View,

Text,

TextInput,

Button,

StyleSheet,

ScrollView,

ActivityIndicator,

Alert,

Linking,

} from 'react-native';

const App = () => {

const [passion, setPassion] = useState('');

const [result, setResult] = useState(null);

const [loading, setLoading] = useState(false);

const [error, setError] = useState(null);

const processPassion = async () => {

setLoading(true);

setError(null);

setResult(null);

try {

*// Simulate API call or data processing*

const processedResult = await fetchPassionData(passion);

setResult(processedResult);

} catch (err) {

setError('Failed to process your passion. Please try again.');

Alert.alert('Error', 'Failed to get results, please try again.');

} finally {

setLoading(false);

}

};

const openLink = (url) => {

Linking.openURL(url);

}

const renderResources = (resources) => {

if (!resources || resources.length === 0) {

return <Text style={styles.resultText}>No resources found.</Text>;

}

return resources.map((resource, index) => (

<View key={index} style={styles.resourceItem}>

<Text style={styles.resourceTitle} onPress={() => openLink(resource.link)}>

{resource.title}

</Text>

<Text style={styles.resourceDescription}>{resource.description}</Text>

</View>

));

};

return (

<ScrollView contentContainerStyle={styles.container}>

<Text style={styles.title}>Enter Your Passion:</Text>

<TextInput

style={styles.input}

placeholder="e.g., Art, Music, Coding"

value={passion}

onChangeText={setPassion}

/>

<Button title="Get Direction" onPress={processPassion} disabled={loading} />

{loading && <ActivityIndicator style={styles.loading} size="large" />}

{error && <Text style={styles.error}>{error}</Text>}

{result && (

<View style={styles.resultContainer}>

<Text style={styles.resultTitle}>Your Direction:</Text>

<Text style={styles.resultText}>{result.direction}</Text>

<Text style={styles.resultTitle}>Resources:</Text>

{renderResources(result.resources)}

</View>

)}

</ScrollView>

);

};

*// ... (styles will be added in the next part)*

export default App;

*// ... (previous code)*

const fetchPassionData = async (passion) => {

*// Simulate API delay*

await new Promise((resolve) => setTimeout(resolve, 1000));

const passionLower = passion.toLowerCase();

if (passionLower.includes('art')) {

return {

direction: 'Explore visual arts, painting, digital art, or graphic design.',

resources: [

{

title: 'Skillshare Art Courses',

description: 'Learn various art techniques from professionals.',

link: 'https://www.skillshare.com/browse/art',

},

{

title: 'Adobe Creative Cloud',

description: 'Tools for digital art and design.',

link: 'https://www.adobe.com/creativecloud.html',

},

],

};

} else if (passionLower.includes('music')) {

return {

direction: 'Learn an instrument, explore music theory, or try songwriting.',

resources: [

{

title: 'Udemy Music Courses',

description: 'Comprehensive music lessons online.',

link: 'https://www.udemy.com/topic/music/',

},

{

title: 'Sheet Music Direct',

description: 'Download and print sheet music.',

link: 'https://www.sheetmusicdirect.com/',

},

],

};

} else if (passionLower.includes('coding')) {

return {

direction: 'Start with web development, app development, or data science.',

resources: [

{

title: 'FreeCodeCamp',

description: 'Learn to code for free.',

link: 'https://www.freecodecamp.org/',

},

{

title: 'GitHub',

description: 'Collaborate on coding projects.',

link: 'https://github.com/',

},

],

};

} else if (passionLower.includes('writing')) {

return {

direction: 'Explore creative writing, journalism, or technical writing.',

resources: [

{

title: 'MasterClass Writing Courses',

description: 'Learn writing from renowned authors.',

link: 'https://www.masterclass.com/categories/writing',

},

{

title: 'The Writer\'s Digest',

description: 'Resources for writers.',

link: 'https://www.writersdigest.com/',

},

],

};

} else {

return {

direction: 'We are still working on your passion. Try another or be more specific.',

resources: [],

};

}

};

*// ... (previous code)*

const styles = StyleSheet.create({

container: {

flexGrow: 1,

padding: 20,

justifyContent: 'center',

},

title: {

fontSize: 20,

marginBottom: 10,

},

input: {

borderWidth: 1,

borderColor: '#ccc',

padding: 10,

marginBottom: 20,

},

resultContainer: {

marginTop: 20,

},

resultTitle: {

fontSize: 18,

fontWeight: 'bold',

marginBottom: 5,

},

resultText: {

marginBottom: 10,

},

loading: {

marginTop: 20,

},

error: {

color: 'red',

marginTop: 10,

},

resourceItem: {

marginBottom: 10,

},

resourceTitle: {

color: 'blue',

textDecorationLine: 'underline',

marginBottom: 5,

},

resourceDescription: {

fontSize: 14,

},

});