WhatsApp 2.0 App UI (Front-end-UI Only) -> A clone of the popular Realtime Chat mobile applications WhatsApp in React Native with WhatsApp UI frontend design. The UI contains a WhatsApp-alike main chat screen. Clicking on these Chats will be redirected to open a private chat room to allow the user to send messages to make conversations with other users. It contains a list of messages with styles and Contacts with statuses.

Developing Languages, Tools, and Technologies Needed:

Expo https://docs.expo.dev/get-started/create-a-new-app/
Xcode iOS 16.1 iPhone 14 Simulator https://developer.apple.com/xcode/
Vscode https://code.visualstudio.com/updates/v1_73
React-Native v0.70.5 https://reactnative.dev/
Day.js 2kB https://day.js.org/
Ether Creative Shadow Generator https://ethercreative.github.io/react-native-shadow-generator/
Expo Icons https://icons.expo.fyi

Prerequisites & Setups:

In local Console, initialize to create the react expo app running:

```
npx create-expo-app WhatsApp2.0
```

Configure the Xcode iOS Simulator:
https://developer.apple.com/forums/thread/678469
Xcode -> Preferences -> locations -> "Command Line Tools"
Open the created project in Vscode and start the server development at Terminal:

```
npm start
```

Press i to open iOS Simulator.
Successfully started Expo Developer Tool Metro Bundler with IOS Simulator.

started expo with ios simulator.PNG

Synchronous Developing Notes:

Image Error: `Image source = {{uri:'...'}}` not displaying image on iOS Bundle.
DEBUGGING: https://github… URL prefix blocked, click Download in github image to obtain a different downloading url. Use CMD+D to inspect elements. SHIFT+ i to switch iOS simulators.

Build the chat list item:

Insert sample chat container, chat content and customize styles in `/ChatListItem/index.js`:

```
  <View style={styles.content}>
                <View style={styles.row}>
                    <Text style={styles.name}>Scarlett</Text>
                    <Text style={styles.subTitle}>8:30</Text>
...
 const styles = StyleSheet.create({
    container: {
        flexDirection: 'row',
        marginHorizontal: 10,
        marginVertical: 5,
        height: 70,
...
```

chat list items created.PNG

Add props in `/ChatListItem/index.js:`

```
const ChatListItem = (props) => {
    console.log(props);
```

fetched props in log.PNG

Import chat list of profile pictures into `ChatsScreen.js:`

```
import { View, Text, FlatList } from 'react-native';
import chats from '../../assets/data/chats.json';
import ChatListItem from '../components/ChatListItem';
const ChatsScreen = () => {
    return (<FlatList data={chats} renderItem={({ item }) =>
<ChatListItem chat={item} />} />
    );
};
export default ChatsScreen;
```

list of chatters profiles showed up.PNG

Remove `align-item: center` in `App.js.`

chats all popped up.PNG

Open another terminal and install Day.js by running:

```
npm install dayjs
```

Build chat screen:

Render messages in `ChatScreen.js:`

```
  <ImageBackground source={bg} style={styles.bg}>
            <FlatList
                data={messages}
                renderItem={({ item }) => <Message message={item} />}
                style={styles.list}
                inverted
            />
        </ImageBackground>
    );
```

text messages are rendered.PNG

render both text contents and time in `/Message/index.js:`

```
     <View style= {styles.container}>
            <Text>{message.text}</Text>
            <Text style={styles.time}>{message.createdAt}</Text>
```

text content and time rendered.PNG

Set colors and flex positions to distinguish from sender and receiver:

```
     <View style={[styles.container, {
            backgroundColor: isMessage() ? '#DCF8C5' : 'white',
            alignSelf: isMessage() ? 'flex-end' : 'flex-start'
```

sender and receiver distinguished.PNG

Add background shadow and rearrange messages order:

```
  //shadows
        shadowColor: '#000',
        shadowOffset: {
            width: 0,
            height: 1,
        },
        shadowOpacity: 0.18,
        shadowRadius: 1.0,
        elevation: 1,
```

modified order and shadow added.PNG

Design text input box and icons in `/InputBox/index.js:`

```
    container: {
        flexDirection: 'row',
        backgroundColor: 'whitesmoke',
        padding: 5,
   },
    input: {
        flex: 1,
        backgroundColor: 'white',
        padding: 5,
        paddingHorizontal: 10,
        borderRadius: 50,
        borderColor: 'lightgray',
        borderWidth: StyleSheet.hairlineWidth,
```

text input box designed.PNG

Set up sending state as console:

```
 //state data
    const [newMessage, setNewMessage] = useState('');
    const onSend = () => {
        console.warn('Sending a new message:', newMessage);
        setNewMessage('');
```

Now sending state console with the sending message content showed when click send button:

sending state console showed.PNG

React navigation:

Install dependencies by running these command in vscode:

```
npx expo install @react-navigation/native
@react-navigation/native-stack @react-navigation/bottom-tabs
react-native-screens react-native-safe-area-context
```

navigation bar displayed.PNG

Import different navigation bars and their icons:

navigation bars imported.PNG

Import different navigation screen bars in `MainTabNavigator.js:`

```
 <Tab.Screen
           name="Status"
               component={NotImplementedScreen}
               options={{
                   tabBarIcon: ({ color, size }) => (
                <Ionicons name="logo-whatsapp" size={size}
color={color} />
...
```

different navigator screens.PNG

Build Contact screen:

Use `onPress` method to be redirected to contacts:

```
 headerRight: () => (
           <Entypo
          onPress={() => navigation.navigate("Contacts")}
                    name="new-message"
                    size={18}
                    color={'royalblue'}
                    style={{ marginRight: 15 }}
```

Now click the button on right-top, contact list showed:

contact list showed.PNG