9. Responsive Styling in React Native

9.1 Create a ProfileCard component styled with StyleSheet. Task: • Apply styles for layout, fonts, and colors using StyleSheet.create(). • Ensure the card looks good on both small and large sc

App.js

import React from 'react';

import ProfileCard from './components/ProfileCard';

function App() {

  return (

    <div>

      <ProfileCard

        name="Nikita Sharma"

        bio="Frontend Developer | AI Explorer"

        imageUrl="https://i.pravatar.cc/150?img=47"

      />

    </div>

  );

}

export default App;

C:\Users\nikit\Desktop\WEB LAB SEM 6\9.1\src\components\ProfileCard.js

import React from 'react';

function getStyles() {

  const width = window.innerWidth;

  const isSmall = width < 600;

  return {

    card: {

      backgroundColor: '#ffffff',

      borderRadius: 10,

      padding: isSmall ? 12 : 20,

      width: isSmall ? '90%' : 300,

      margin: '20px auto',

      boxShadow: '0 4px 8px rgba(0,0,0,0.1)',

      textAlign: 'center',

    },

    image: {

      width: isSmall ? 100 : 150,

      height: isSmall ? 100 : 150,

      borderRadius: '50%',

      objectFit: 'cover',

      marginBottom: 12,

    },

    name: {

      fontSize: isSmall ? 18 : 22,

      fontWeight: 'bold',

      marginBottom: 6,

    },

    bio: {

      fontSize: isSmall ? 14 : 16,

      color: '#555',

    },

  };

}

const ProfileCard = ({ name, bio, imageUrl }) => {

  const styles = getStyles();

  return (

    <div style={styles.card}>

      <img src={imageUrl} alt="profile" style={styles.image} />

      <div style={styles.name}>{name}</div>

      <div style={styles.bio}>{bio}</div>

    </div>

  );

};

export default ProfileCard;

9.2 Design a login screen that adjusts layout on different device sizes. Task: • Use StyleSheet to style inputs and buttons. • Implement media query-like behavior by detecting screen width and modifying layout (e.g., stacking inputs vertically on small screens and horizontally on tablets).

App.js

import React from 'react';

import LoginScreen from './components/LoginScreen';

function App() {

  return (

    <div>

      <LoginScreen />

    </div>

  );

}

export default App;

C:\Users\nikit\Desktop\WEB LAB SEM 6\9.2\src\components\LoginScreen.js

import React, { useState, useEffect } from 'react';

function getStyles(isSmallScreen) {

  return {

    container: {

      display: 'flex',

      flexDirection: 'column',

      alignItems: 'center',

      padding: 20,

    },

    form: {

      display: 'flex',

      flexDirection: isSmallScreen ? 'column' : 'row',

      gap: 15,

      alignItems: 'center',

      justifyContent: 'center',

      marginTop: 20,

    },

    input: {

      padding: 10,

      fontSize: 16,

      borderRadius: 5,

      border: '1px solid #ccc',

      width: isSmallScreen ? '90%' : 200,

    },

    button: {

      padding: 10,

      fontSize: 16,

      borderRadius: 5,

      backgroundColor: '#007bff',

      color: '#fff',

      border: 'none',

      cursor: 'pointer',

      width: isSmallScreen ? '90%' : 120,

    },

  };

}

const LoginScreen = () => {

  const [isSmallScreen, setIsSmallScreen] = useState(window.innerWidth < 600);

  const [email, setEmail] = useState('');

  const [password, setPassword] = useState('');

  // Update on window resize

  useEffect(() => {

    const handleResize = () => setIsSmallScreen(window.innerWidth < 600);

    window.addEventListener('resize', handleResize);

    return () => window.removeEventListener('resize', handleResize);

  }, []);

  const styles = getStyles(isSmallScreen);

  const handleLogin = () => {

    alert(`Email: ${email}, Password: ${password}`);

  };

  return (

    <div style={styles.container}>

      <h2>Login</h2>

      <div style={styles.form}>

        <input

          type="email"

          placeholder="Email"

          style={styles.input}

          value={email}

          onChange={(e) => setEmail(e.target.value)}

        />

        <input

          type="password"

          placeholder="Password"

          style={styles.input}

          value={password}

          onChange={(e) => setPassword(e.target.value)}

        />

        <button onClick={handleLogin} style={styles.button}>

          Log In

        </button>

      </div>

    </div>

  );

};

export default LoginScreen;