```
// src/screens/guides/GuideDetailScreen.js (continued)
          {guide.steps.map((step, index) => (
            <List.Item
              key={step.id}
              title={`${index + 1}. ${step.title}`}
              description={step.description}
              left={props => <List.Icon {...props} icon="check-circle"
color="#4f8ef7" />}
              style={styles.stepItem}
            />
          ))}

          <View style={styles.buttonContainer}>
            <Button
              mode="contained"
              icon="play"
              onPress={() => navigation.navigate('GuideStep', {
                guideId: guide.id,
                title: guide.title,
                steps: guide.steps,
                currentStep: 0
              })}
              style={styles.startButton}
            >
              Start Guide
            </Button>
          </View>
        </View>
      </ScrollView>
    </SafeAreaView>
  );
};

const styles = StyleSheet.create({
  container: {
    flex: 1,
    backgroundColor: '#f5f5f5',
  },
  headerCard: {
    margin: 16,
    borderRadius: 10,
  },
  title: {
    fontSize: 22,
    fontWeight: 'bold',
    marginBottom: 10,
  },
```

```
  metaRow: {
    flexDirection: 'row',
    marginBottom: 15,
    flexWrap: 'wrap',
  },
  metaItem: {
    flexDirection: 'row',
    alignItems: 'center',
    marginRight: 15,
    marginBottom: 5,
  },
  metaText: {
    marginLeft: 5,
    color: '#666',
  },
  description: {
    fontSize: 16,
    lineHeight: 24,
    marginBottom: 15,
    color: '#333',
  },
  image: {
    width: '100%',
    height: 200,
    borderRadius: 10,
    marginBottom: 15,
  },
  createdBy: {
    fontSize: 14,
    color: '#888',
    fontStyle: 'italic',
  },
  section: {
    padding: 16,
  },
  sectionTitle: {
    fontSize: 18,
    fontWeight: 'bold',
    marginBottom: 15,
  },
  stepItem: {
    backgroundColor: 'white',
    marginBottom: 8,
    borderRadius: 8,
  },
  buttonContainer: {
    marginTop: 20,
```

```
    alignItems: 'center',
  },
  startButton: {
    width: '80%',
    paddingVertical: 8,
    backgroundColor: '#4f8ef7',
  },
});

export default GuideDetailScreen;

// src/screens/guides/GuideStepScreen.js
import React, { useState } from 'react';
import { View, Text, StyleSheet, Image, ScrollView, TouchableOpacity }
from 'react-native';
import { Button, ProgressBar, IconButton } from 'react-native-paper';
import { Ionicons } from '@expo/vector-icons';
import { SafeAreaView } from 'react-native-safe-area-context';
import * as Speech from 'expo-speech';

const GuideStepScreen = ({ route, navigation }) => {
  const { guideId, title, steps, currentStep: initialStep } =
route.params;
  const [currentStep, setCurrentStep] = useState(initialStep);
  const [isSpeaking, setIsSpeaking] = useState(false);

  const step = steps[currentStep];
  const progress = (currentStep + 1) / steps.length;

  const speakStep = () => {
    if (isSpeaking) {
      Speech.stop();
      setIsSpeaking(false);
      return;
    }

    const textToSpeak = `Step ${currentStep + 1}: ${step.title}.
${step.description}`;
    Speech.speak(textToSpeak, {
      language: 'en',
      onDone: () => setIsSpeaking(false),
    });
    setIsSpeaking(true);
  };

  const goToPreviousStep = () => {
    if (currentStep > 0) {
```

```
      Speech.stop();
      setIsSpeaking(false);
      setCurrentStep(currentStep - 1);
    }
  };

  const goToNextStep = () => {
    if (currentStep < steps.length - 1) {
      Speech.stop();
      setIsSpeaking(false);
      setCurrentStep(currentStep + 1);
    } else {
      // This is the last step, go back to guide detail
      navigation.navigate('GuideDetail', { id: guideId, title });
    }
  };

  return (
    <SafeAreaView style={styles.container}>
      <View style={styles.progressContainer}>
        <Text style={styles.progressText}>
          Step {currentStep + 1} of {steps.length}
        </Text>
        <ProgressBar progress={progress} color="#4f8ef7"
style={styles.progressBar} />
      </View>

      <ScrollView style={styles.contentContainer}>
        <Text style={styles.stepTitle}>{step.title}</Text>

        {/* Step image would go here */}
        <View style={styles.imageContainer}>
          <Image
            source={{ uri: 'https://via.placeholder.com/350x200' }}
            style={styles.stepImage}
            resizeMode="contain"
          />
        </View>

        <Text style={styles.stepDescription}>{step.description}</Text>

        <View style={styles.audioButtonContainer}>
          <TouchableOpacity style={styles.audioButton}
onPress={speakStep}>
            <Ionicons name={isSpeaking ? 'stop' : 'volume-high'} size={24}
color="white" />
            <Text style={styles.audioButtonText}>
```

```
              {isSpeaking ? 'Stop Reading' : 'Read Aloud'}
            </Text>
          </TouchableOpacity>
        </View>

        <View style={styles.helpBox}>
          <Text style={styles.helpTitle}>Need more help?</Text>
          <Button
            mode="outlined"
            icon="video"
            onPress={() => navigation.navigate('Video Call')}
            style={styles.videoCallButton}
          >
            Start Video Call with Helper
          </Button>
        </View>
      </ScrollView>

      <View style={styles.navigationContainer}>
        <IconButton
          icon="chevron-left"
          size={30}
          onPress={goToPreviousStep}
          disabled={currentStep === 0}
          style={[styles.navButton, currentStep === 0 &&
styles.disabledButton]}
        />
        <Button
          mode="contained"
          onPress={goToNextStep}
          style={styles.nextButton}
        >
          {currentStep < steps.length - 1 ? 'Next Step' : 'Finish'}
        </Button>
        <IconButton
          icon="chevron-right"
          size={30}
          onPress={goToNextStep}
          style={styles.navButton}
        />
      </View>
    </SafeAreaView>
  );
};

const styles = StyleSheet.create({
  container: {
```

```
    flex: 1,
    backgroundColor: '#f5f5f5',
  },
  progressContainer: {
    padding: 16,
    backgroundColor: 'white',
  },
  progressText: {
    fontSize: 14,
    marginBottom: 5,
    textAlign: 'center',
  },
  progressBar: {
    height: 6,
    borderRadius: 3,
  },
  contentContainer: {
    flex: 1,
    padding: 16,
  },
  stepTitle: {
    fontSize: 22,
    fontWeight: 'bold',
    marginBottom: 20,
    textAlign: 'center',
  },
  imageContainer: {
    alignItems: 'center',
    marginBottom: 20,
  },
  stepImage: {
    width: '100%',
    height: 200,
    borderRadius: 10,
  },
  stepDescription: {
    fontSize: 18,
    lineHeight: 28,
    marginBottom: 20,
  },
  audioButtonContainer: {
    alignItems: 'center',
    marginBottom: 20,
  },
  audioButton: {
    flexDirection: 'row',
    alignItems: 'center',
```

```
    backgroundColor: '#4f8ef7',
    paddingVertical: 10,
    paddingHorizontal: 20,
    borderRadius: 25,
  },
  audioButtonText: {
    color: 'white',
    marginLeft: 10,
    fontWeight: '500',
  },
  helpBox: {
    backgroundColor: '#f0f7ff',
    padding: 16,
    borderRadius: 10,
    marginBottom: 20,
  },
  helpTitle: {
    fontSize: 16,
    fontWeight: 'bold',
    marginBottom: 10,
  },
  videoCallButton: {
    borderColor: '#4f8ef7',
    borderWidth: 1,
  },
  navigationContainer: {
    flexDirection: 'row',
    justifyContent: 'space-between',
    alignItems: 'center',
    padding: 16,
    backgroundColor: 'white',
    borderTopWidth: 1,
    borderTopColor: '#eee',
  },
  navButton: {
    backgroundColor: '#f0f7ff',
  },
  disabledButton: {
    opacity: 0.5,
  },
  nextButton: {
    flex: 1,
    marginHorizontal: 10,
    backgroundColor: '#4f8ef7',
  },
});
```

```
export default GuideStepScreen;

// src/screens/guides/CreateGuideScreen.js
import React, { useState } from 'react';
import { View, Text, StyleSheet, ScrollView, TextInput, Alert } from
'react-native';
import { Button, IconButton, Chip, HelperText } from 'react-native-paper';
import { SafeAreaView } from 'react-native-safe-area-context';

const CreateGuideScreen = ({ navigation }) => {
  const [title, setTitle] = useState('');
  const [category, setCategory] = useState('');
  const [description, setDescription] = useState('');
  const [steps, setSteps] = useState([
    { id: 1, title: '', description: '' },
  ]);

  const categories = ['Email', 'Social Media', 'Internet', 'Smartphone',
'Computer', 'Other'];

  const addStep = () => {
    const newStep = {
      id: steps.length + 1,
      title: '',
      description: '',
    };
    setSteps([...steps, newStep]);
  };

  const removeStep = (idToRemove) => {
    if (steps.length <= 1) {
      Alert.alert('Error', 'A guide must have at least one step.');
      return;
    }

    const updatedSteps = steps.filter(step => step.id !==
idToRemove).map((step, index) => ({
      ...step,
      id: index + 1, // Reindex the steps
    }));

    setSteps(updatedSteps);
  };

  const updateStep = (id, field, value) => {
    const updatedSteps = steps.map(step => {
      if (step.id === id) {
```

```
        return { ...step, [field]: value };
      }
      return step;
    });

    setSteps(updatedSteps);
  };

  const saveGuide = () => {
    // Validate inputs
    if (!title.trim()) {
      Alert.alert('Error', 'Please provide a title for your guide.');
      return;
    }

    if (!category) {
      Alert.alert('Error', 'Please select a category for your guide.');
      return;
    }

    const hasEmptySteps = steps.some(step => !step.title.trim() ||
!step.description.trim());
    if (hasEmptySteps) {
      Alert.alert('Error', 'Please complete all steps with titles and
descriptions.');
      return;
    }

    // In a real app, we would save the guide to a database or local
storage
    Alert.alert('Success', 'Your guide has been created!', [
      { text: 'OK', onPress: () => navigation.goBack() }
    ]);
  };

  return (
    <SafeAreaView style={styles.container}>
      <ScrollView style={styles.content}>
        <Text style={styles.sectionTitle}>Guide Information</Text>

        <View style={styles.inputContainer}>
          <Text style={styles.label}>Title</Text>
          <TextInput
            style={styles.input}
            placeholder="Enter guide title"
            value={title}
            onChangeText={setTitle}
```

```
          />
        </View>

        <View style={styles.inputContainer}>
          <Text style={styles.label}>Category</Text>
          <View style={styles.categoriesContainer}>
            {categories.map((cat) => (
              <Chip
                key={cat}
                selected={category === cat}
                onPress={() => setCategory(cat)}
                style={[
                  styles.categoryChip,
                  category === cat && styles.selectedCategoryChip
                ]}
                textStyle={[
                  styles.categoryChipText,
                  category === cat && styles.selectedCategoryChipText
                ]}
              >
                {cat}
              </Chip>
            ))}
          </View>
        </View>

        <View style={styles.inputContainer}>
          <Text style={styles.label}>Description</Text>
          <TextInput
            style={[styles.input, styles.textArea]}
            placeholder="Enter guide description"
            value={description}
            onChangeText={setDescription}
            multiline
            numberOfLines={4}
          />
        </View>

        <Text style={styles.sectionTitle}>Guide Steps</Text>
        <HelperText>Create step-by-step instructions for your
guide</HelperText>

        {steps.map((step) => (
          <View key={step.id} style={styles.stepContainer}>
            <View style={styles.stepHeader}>
              <Text style={styles.stepTitle}>Step {step.id}</Text>
              <IconButton
```

```
                icon="close"
                size={20}
                onPress={() => removeStep(step.id)}
              />
            </View>

            <View style={styles.inputContainer}>
              <Text style={styles.label}>Title</Text>
              <TextInput
                style={styles.input}
                placeholder={`Enter title for step ${step.id}`}
                value={step.title}
                onChangeText={(text) => updateStep(step.id, 'title',
text)}
              />
            </View>

            <View style={styles.inputContainer}>
              <Text style={styles.label}>Description</Text>
              <TextInput
                style={[styles.input, styles.textArea]}
                placeholder={`Enter description for step ${step.id}`}
                value={step.description}
                onChangeText={(text) => updateStep(step.id, 'description',
text)}
                multiline
                numberOfLines={3}
              />
            </View>
          </View>
        ))}

        <Button
          mode="outlined"
          icon="plus"
          onPress={addStep}
          style={styles.addStepButton}
        >
          Add Step
        </Button>

        <Button
          mode="contained"
          onPress={saveGuide}
          style={styles.saveButton}
        >
          Save Guide
```

```
        </Button>
      </ScrollView>
    </SafeAreaView>
  );
};

const styles = StyleSheet.create({
  container: {
    flex: 1,
    backgroundColor: '#f5f5f5',
  },
  content: {
    padding: 16,
  },
  sectionTitle: {
    fontSize: 18,
    fontWeight: 'bold',
    marginBottom: 15,
    marginTop: 10,
  },
  inputContainer: {
    marginBottom: 15,
  },
  label: {
    fontSize: 16,
    marginBottom: 5,
    color: '#555',
  },
  input: {
    backgroundColor: 'white',
    padding: 12,
    borderRadius: 8,
    borderWidth: 1,
    borderColor: '#ddd',
  },
  textArea: {
    minHeight: 100,
    textAlignVertical: 'top',
  },
  categoriesContainer: {
    flexDirection: 'row',
    flexWrap: 'wrap',
  },
  categoryChip: {
    margin: 4,
    backgroundColor: 'white',
  },
```

```
  selectedCategoryChip: {
    backgroundColor: '#4f8ef7',
  },
  categoryChipText: {
    color: '#333',
  },
  selectedCategoryChipText: {
    color: 'white',
  },
  stepContainer: {
    backgroundColor: 'white',
    borderRadius: 8,
    padding: 15,
    marginBottom: 15,
    borderWidth: 1,
    borderColor: '#eee',
  },
  stepHeader: {
    flexDirection: 'row',
    justifyContent: 'space-between',
    alignItems: 'center',
    marginBottom: 10,
  },
  stepTitle: {
    fontSize: 16,
    fontWeight: 'bold',
  },
  addStepButton: {
    marginTop: 10,
    marginBottom: 20,
    borderColor: '#4f8ef7',
  },
  saveButton: {
    marginBottom: 30,
    paddingVertical: 8,
    backgroundColor: '#4f8ef7',
  },
});

export default CreateGuideScreen;

// src/screens/videocall/VideoCallHomeScreen.js
import React, { useState } from 'react';
import { View, Text, StyleSheet, ScrollView, TouchableOpacity, Image }
from 'react-native';
import { Card, Button, Badge, Divider, Title, Paragraph } from
'react-native-paper';
```

```
import { Ionicons } from '@expo/vector-icons';
import { SafeAreaView } from 'react-native-safe-area-context';

const VideoCallHomeScreen = ({ navigation }) => {
  const [favoriteHelpers, setFavoriteHelpers] = useState([
    { id: 1, name: 'Sarah Johnson', availability: 'Available', rating:
4.9, specialties: ['Email', 'Social Media'], imageUrl:
'https://randomuser.me/api/portraits/women/44.jpg' },
    { id: 2, name: 'Michael Chen', availability: 'Busy', rating: 4.8,
specialties: ['Smartphone', 'Computer'], imageUrl:
'https://randomuser.me/api/portraits/men/32.jpg' },
  ]);

  return (
    <SafeAreaView style={styles.container}>
      <ScrollView>
        <Card style={styles.headerCard}>
          <Card.Content>
            <Title style={styles.headerTitle}>Get Live Help Now</Title>
            <Paragraph style={styles.headerParagraph}>
              Connect with a friendly tech expert via video call for
personalized assistance
            </Paragraph>
            <Button
              mode="contained"
              icon="video"
              onPress={() => navigation.navigate('FindHelper')}
              style={styles.findHelperButton}
            >
              Find Available Helper
            </Button>
          </Card.Content>
        </Card>

        <View style={styles.section}>
          <Text style={styles.sectionTitle}>Your Favorite Helpers</Text>
          {favoriteHelpers.map(helper => (
            <TouchableOpacity
              key={helper.id}
              onPress={() => navigation.navigate('HelperProfile', { id:
helper.id, name: helper.name })}
            >
              <Card style={styles.helperCard}>
                <Card.Content style={styles.helperCardContent}>
                  <Image
                    source={{ uri: helper.imageUrl }}
                    style={styles.helperImage}
```

```
                    />
                  <View style={styles.helperInfo}>
                    <Text style={styles.helperName}>{helper.name}</Text>
                    <View style={styles.helperRating}>
                      <Ionicons name="star" size={16} color="#FFD700" />
                      <Text
style={styles.ratingText}>{helper.rating}</Text>
                    </View>
                    <View style={styles.specialtiesContainer}>
                      {helper.specialties.map((specialty, index) => (
                        <Text key={index} style={styles.specialtyText}>
                          {specialty}{index < helper.specialties.length -
1 ? ', ' : ''}
                        </Text>
                      ))}
                    </View>
                  </View>
                  <View style={styles.availabilityContainer}>
                    <Badge
                      style={[
                        styles.availabilityBadge,
                        { backgroundColor: helper.availability ===
'Available' ? '#4CAF50' : '#FFA000' }
                      ]}
                    >
                      {helper.availability}
                    </Badge>
                    {helper.availability === 'Available' && (
                      <Button
                        mode="outlined"
                        onPress={() => navigation.navigate('Call', {
helperId: helper.id })}
                        style={styles.callButton}
                      >
                        Call
                      </Button>
                    )}
                  </View>
                </Card.Content>
              </Card>
            </TouchableOpacity>
          ))}
        </View>

        <View style={styles.section}>
          <Text style={styles.sectionTitle}>How It Works</Text>
          <Card style={styles.instructionCard}>
```

```
            <Card.Content>
              <View style={styles.instructionStep}>
                <View style={styles.instructionIcon}>
                  <Ionicons name="search" size={24} color="#4f8ef7" />
                </View>
                <View style={styles.instructionText}>
                  <Text style={styles.instructionTitle}>1. Find a
Helper</Text>
                  <Text style={styles.instructionDescription}>
                    Browse available helpers with expertise in your area
of need
                  </Text>
                </View>
              </View>

              <Divider style={styles.divider} />

              <View style={styles.instructionStep}>
                <View style={styles.instructionIcon}>
                  <Ionicons name="videocam" size={24} color="#4f8ef7" />
                </View>
                <View style={styles.instructionText}>
                  <Text style={styles.instructionTitle}>2. Connect via
Video</Text>
                  <Text style={styles.instructionDescription}>
                    Start a secure video call with your chosen helper
                  </Text>
                </View>
              </View>

              <Divider style={styles.divider} />

              <View style={styles.instructionStep}>
                <View style={styles.instructionIcon}>
                  <Ionicons name="help-buoy" size={24} color="#4f8ef7" />
                </View>
                <View style={styles.instructionText}>
                  <Text style={styles.instructionTitle}>3. Get
Support</Text>
                  <Text style={styles.instructionDescription}>
                    Show your helper what you need assistance with and
follow their guidance
                  </Text>
                </View>
              </View>
            </Card.Content>
          </Card>
```

```
        </View>

        <View style={styles.section}>
          <Text style={styles.sectionTitle}>Need Help With</Text>
          <View style={styles.topicsGrid}>
            {['Email', 'Internet', 'Smartphone', 'Computer', 'Social
Media', 'Video Calls'].map((topic, index) => (
              <TouchableOpacity
                key={index}
                style={styles.topicButton}
                onPress={() => navigation.navigate('FindHelper', { filter:
topic })}
              >
                <View style={styles.topicIcon}>
                  <Ionicons
                    name={
                      topic === 'Email' ? 'mail' :
                      topic === 'Internet' ? 'globe' :
                      topic === 'Smartphone' ? 'phone-portrait' :
                      topic === 'Computer' ? 'desktop' :
                      topic === 'Social Media' ? 'people' :
                      'videocam'
                    }
                    size={24}
                    color="#4f8ef7"
                  />
                </View>
                <Text style={styles.topicText}>{topic}</Text>
              </TouchableOpacity>
            ))}
          </View>
        </View>
      </ScrollView>
    </SafeAreaView>
  );
};

const styles = StyleSheet.create({
  container: {
    flex: 1,
    backgroundColor: '#f5f5f5',
  },
  headerCard: {
    margin: 16,
    borderRadius: 10,
    backgroundColor: '#4f8ef7',
  },
```

```
  headerTitle: {
    color: 'white',
    fontSize: 22,
    fontWeight: 'bold',
  },
  headerParagraph: {
    color: 'white',
    fontSize: 16,
    marginBottom: 20,
  },
  findHelperButton: {
    backgroundColor: 'white',
    marginTop: 10,
  },
  section: {
    padding: 16,
  },
  sectionTitle: {
    fontSize: 18,
    fontWeight: 'bold',
    marginBottom: 15,
  },
  helperCard: {
    marginBottom: 12,
    borderRadius: 10,
  },
  helperCardContent: {
    flexDirection: 'row',
    alignItems: 'center',
  },
  helperImage: {
    width: 60,
    height: 60,
    borderRadius: 30,
    marginRight: 15,
  },
  helperInfo: {
    flex: 1,
  },
  helperName: {
    fontSize: 16,
    fontWeight: 'bold',
    marginBottom: 4,
  },
  helperRating: {
    flexDirection: 'row',
    alignItems: 'center',
```

```
    marginBottom: 4,
  },
  ratingText: {
    marginLeft: 4,
    fontSize: 14,
    color: '#666',
  },
  specialtiesContainer: {
    flexDirection: 'row',
    flexWrap: 'wrap',
  },
  specialtyText: {
    fontSize: 12,
    color: '#666',
  },
  availabilityContainer: {
    alignItems: 'center',
  },
  availabilityBadge: {
    marginBottom: 8,
  },
  callButton: {
    height: 32,
    borderColor: '#4f8ef7',
  },
  instructionCard: {
    borderRadius: 10,
    elevation: 2,
  },
  instructionStep: {
    flexDirection: 'row',
    alignItems: 'center',
    paddingVertical: 15,
  },
  instructionIcon: {
    width: 50,
    height: 50,
    borderRadius: 25,
    backgroundColor: '#e6f0ff',
    justifyContent: 'center',
    alignItems: 'center',
    marginRight: 15,
  },
  instructionText: {
    flex: 1,
  },
  instructionTitle: {
```

```
    fontSize: 16,
    fontWeight: 'bold',
    marginBottom: 4,
  },
  instructionDescription: {
    fontSize: 14,
    color: '#666',
    lineHeight: 20,
  },
  divider: {
    height: 1,
    backgroundColor: '#eee',
  },
  topicsGrid: {
    flexDirection: 'row',
    flexWrap: 'wrap',
    justifyContent: 'space-between',
  },
  topicButton: {
    width: '30%',
    backgroundColor: 'white',
    borderRadius: 10,
    padding: 15,
    marginBottom: 12,
    alignItems: 'center',
  },
  topicIcon: {
    width: 50,
    height: 50,
    borderRadius: 25,
    backgroundColor: '#e6f0ff',
    justifyContent: 'center',
    alignItems: 'center',
    marginBottom: 10,
  },
  topicText: {
    fontSize: 14,
    textAlign: 'center',
  },
});

export default VideoCallHomeScreen;

// src/screens/videocall/FindHelperScreen.js
import React, { useState, useEffect } from 'react';
import { View, Text, StyleSheet, FlatList, Image, TouchableOpacity } from
'react-native';
```

```
import { Searchbar, Chip, Button, Badge, Card, ActivityIndicator } from
'react-native-paper';
import { Ionicons } from '@expo/vector-icons';
import { SafeAreaView } from 'react-native-safe-area-context';

const FindHelperScreen = ({ route, navigation }) => {
  const [searchQuery, setSearchQuery] = useState('');
  const [selectedCategory, setSelectedCategory] =
useState(route.params?.filter || null);
  const [helpers, setHelpers] = useState([]);
  const [loading, setLoading] = useState(true);

  const categories = [
    { id: 1, name: 'All', icon: 'apps' },
    { id: 2, name: 'Email', icon: 'mail' },
    { id: 3, name: 'Social Media', icon: 'people' },
    { id: 4, name: 'Internet', icon: 'globe' },
    { id: 5, name: 'Smartphone', icon: 'phone-portrait' },
    { id: 6, name: 'Computer', icon: 'desktop' },
  ];

  useEffect(() => {
    // Simulating data fetch with a timeout
    setTimeout(() => {
      const mockHelpers = [
        {
          id: 1,
          name: 'Sarah Johnson',
          availability: 'Available',
          rating: 4.9,
          price: 'Free',
          specialties: ['Email', 'Social Media'],
          description: 'Friendly helper with 10+ years of teaching seniors
about technology.',
          imageUrl: 'https://randomuser.me/api/portraits/women/44.jpg'
        },
        {
          id: 2,
          name: 'Michael Chen',
          availability: 'Available',
          rating: 4.8,
          price: 'Free',
          specialties: ['Smartphone', 'Computer'],
          description: 'Patient and clear instructor for all smartphone
and computer questions.',
          imageUrl: 'https://randomuser.me/api/portraits/men/32.jpg'
        },
```

```
        {
            id: 3,
            name: 'Lisa Rodriguez',
            availability: 'Available',
            rating: 4.7,
            price: 'Free',
            specialties: ['Internet', 'Email'],
            description: 'Experienced in helping seniors navigate the
internet safely.',
            imageUrl: 'https://randomuser.me/api/portraits/women/58.jpg'
        },
        {
            id: 4,
            name: 'Robert Williams',
            availability: 'Busy',
            rating: 4.9,
            price: 'Free
```