NAME : SOHAM SHETYE ROLL NO. : 54
DIV : D15A BATCH : C

MAD PWA LAB

LAB 3

Aim: To include icon images and fonts in flutter.

THEORY :

Custom Fonts in Flutter:

What is a Font?
A font is a set of characters with a specific style and size. Fonts are used to define the visual appearance of text in your application.

Custom Fonts in Flutter:
In Flutter, you can use custom fonts to give a unique look to the text in your application. Custom fonts are often used to match a specific brand or design aesthetic.

- How to Add Custom Fonts:
  1. Font Files: Custom fonts are typically provided as font files (e.g., TrueType Font - `.ttf` or OpenType Font - `.otf`).
  2. Directory: Place your font files in the `assets/fonts/` directory in your Flutter project.
  3. Update `pubspec.yaml`: Add an entry in your `pubspec.yaml` file specifying the font file and its family name.

Example:
```yaml
flutter:
  fonts:
    - family: YourCustomFont
      fonts:
        - asset: assets/fonts/your_custom_font.ttf
```

- Using Custom Fonts in Text Widget:
  Use the `fontFamily` property in the `TextStyle` to apply your custom font to a `Text` widget.

Example:
```dart
Text(
  'Hello, World!',
  style: TextStyle(fontFamily: 'YourCustomFont'),
)
```

Images in Flutter:

What is an Image?
An image is a visual representation or graphic in a digital form. In Flutter, images are used to display visual content.

Images in Flutter:
Flutter supports different types of images, including local assets and network images.

- How to Add Images:
  1. Image Files: Images can be PNG, JPEG, GIF, etc.
  2. Directory:*Place your image files in the `assets/images/` directory in your Flutter project.
  3. Update `pubspec.yaml`:** Add an entry to the `assets` section in your `pubspec.yaml` file to include the image files.

Example:
```yaml
flutter:
  assets:
    - assets/images/your_image.png
```

- Using Images in Flutter:
  You can use the `Image` widget or `Image.asset` to display images.

Example:
```dart
Image.asset('assets/images/your_image.png')
```

```

Icons in Flutter:

What is an Icon?
An icon is a visual symbol representing an object, concept, or action. Icons are used to provide a quick and recognizable representation of functionality in an application.

Icons in Flutter:
Flutter provides a set of built-in icons that you can use directly. Additionally, you can use custom icons by importing image assets or using custom icon font libraries.

- Using Built-in Icons:
  Flutter's `Icons` class provides a variety of built-in icons that you can use directly.

Example:
```dart
Icon(Icons.home)
```

- Using Custom Icons:
  1. Image Assets: Import custom icons as image assets.

  Example:
  ```dart
  Image.asset('assets/icons/custom_icon.png')
  ```

  2. Icon Fonts: Use custom icon fonts (e.g., Font Awesome) by importing and using the corresponding icons.

  Example:
  ```dart
  Icon(FontAwesomeIcons.someIcon)
  ```

These concepts collectively contribute to the visual design and aesthetics of your Flutter application, allowing you to create a visually appealing and brand-consistent user interface.
  ```

CODE :

```
import 'package:flutter/material.dart';

void main() {
  runApp(MyApp());
}

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: PhoneNumberInputScreen(),
    );
  }
}

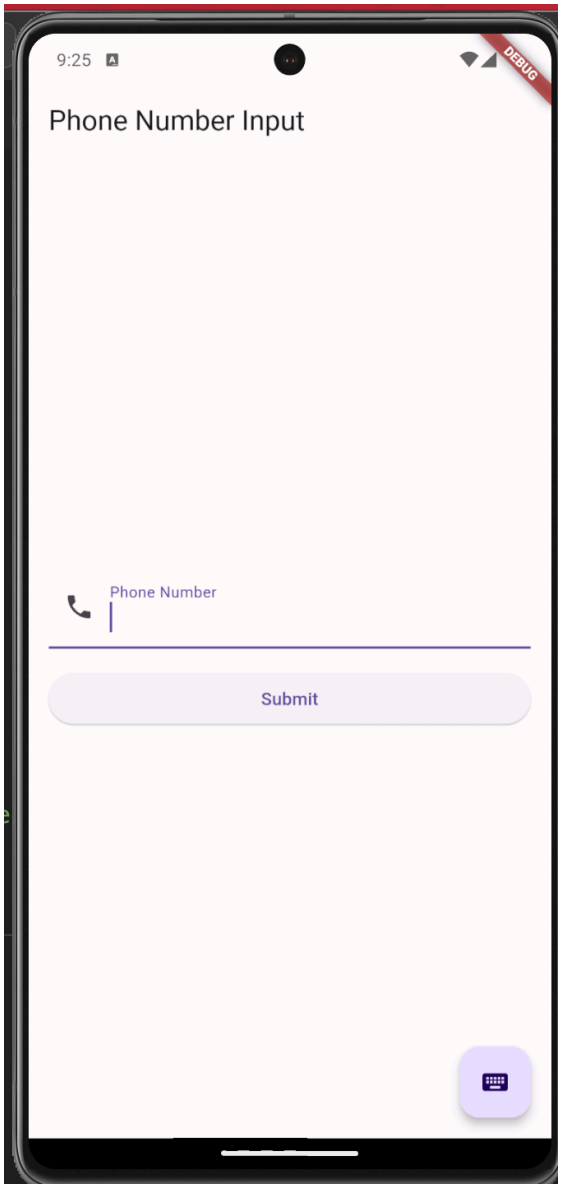
class PhoneNumberInputScreen extends StatefulWidget {
  @override
  _PhoneNumberInputScreenState createState() => _PhoneNumberInputScreenState();
}

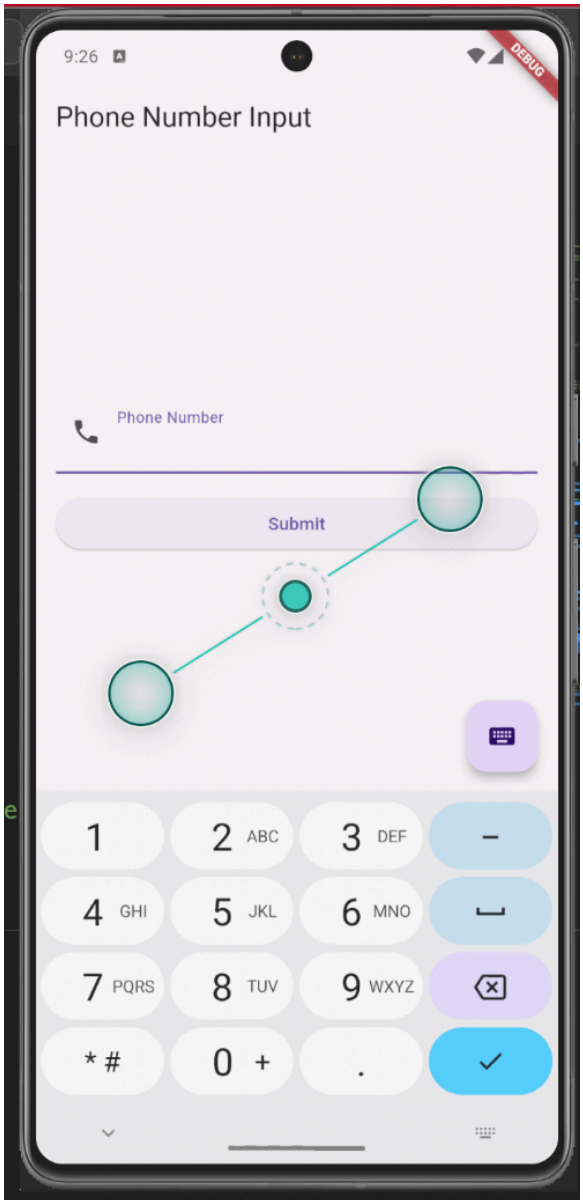
class _PhoneNumberInputScreenState extends State<PhoneNumberInputScreen> {
  TextEditingController _phoneNumberController = TextEditingController();

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Enter Phone number for verification'),
      ),
      body: Padding(
        padding: const EdgeInsets.all(16.0),
        child: Column(
          mainAxisAlignment: MainAxisAlignment.center,
          crossAxisAlignment: CrossAxisAlignment.stretch,
          children: [
            TextField(
              controller: _phoneNumberController,
              keyboardType: TextInputType.phone,
              decoration: InputDecoration(
```

```
            labelText: 'Phone Number',
            prefixIcon: Icon(Icons.phone),
          ),
        ),
        SizedBox(height: 16.0),
        ElevatedButton(
          onPressed: () {
            // Handle phone number submission
            String phoneNumber = _phoneNumberController.text;
            print('Phone Number: $phoneNumber');
          },
          child: Text('Submit'),
        ),
      ],
    ),
  ),
  floatingActionButton: FloatingActionButton(
    onPressed: () {
      // Show floating keypad
      // You can implement your own keypad logic or use a package
      // like 'flutter_numeric_keyboard' for a numeric keypad.
      // For simplicity, we'll just print a message here.
      print('Floating Keypad Pressed');
    },
    tooltip: 'Floating Keypad',
    child: Icon(Icons.keyboard),
  ),
 );
 }
}
```

OUTPUT :


9:25
DEBUG
Phone Number Input
Phone Number
Submit



The code uses built-in Flutter icons from the Icons class.

- Icons.person for the name input.
- Icons.email for the email input.
- Icons.phone for the phone number input.
- Icons.lock for the password input.
- ppBar: The app bar in the PhoneNumberInputScreen uses the Icons.phone icon for the prefix of the phone number input.
- ElevatedButton: The submit button uses the default Icons.check icon.
- FloatingActionButton: The floating action button (FAB) uses the Icons.keyboard icon.

- AppBar Action: An additional app bar action is included with the Icons.info icon, which is triggered when the user presses the info button.

Conclusion :
The provided Flutter registration page code utilizes built-in Flutter icons (`Icons.person`, `Icons.email`, `Icons.phone`, and `Icons.lock`) for user input fields, and it doesn't include specific images or custom fonts, allowing flexibility for customization based on individual design preferences.