Aim: To create an interactive Form using a form widget.

Theory:
In Flutter, forms are a fundamental part of building user interfaces, allowing users to input data and interact with your application. The TextFormField widget is commonly used to create text input fields within a form.
Components of Form:
1. Form: A Form widget is a container for a group of form fields (such as TextFormField, Checkbox, Radio, etc.). It manages the form's state, validation, and submission.

2. TextFormField: A TextFormField widget is used to create a text input field within a form. It provides several properties for customizing the appearance and behavior of the input field, such as decoration (for styling), validator (for input validation), onSaved (for saving the input value), and more.

3. Button: In the context of forms, buttons are used to submit or reset the form. You can use the ElevatedButton, TextButton, OutlinedButton, or IconButton widgets to create buttons in Flutter. Typically, a button's onPressed callback is used to handle form submission or other actions.

4. Validator: A validator is a function that determines whether the input provided by the user is valid. It is commonly used with form fields like TextFormField to perform validation before submitting the form. The validator function takes the input value as a parameter and returns an error message if the input is invalid, or null if the input is valid.

5. GlobalKey<FormState>: This is a key that uniquely identifies the form. It's used to interact with the form's state and validate the form. In this code, it's assigned to the _formKey variable.

6. ElevatedButton: This is a button widget that is typically used for actions in the user interface. In this case, it's used for the "Create Invoice" button, which triggers the submission of the form.

Code:

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

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

```dart
class MyApp extends StatelessWidget {
 @override
 Widget build(BuildContext context) {
   return MaterialApp(
     title: 'ERP Application',
     theme: ThemeData(
       primarySwatch: Colors.blue,
       visualDensity: VisualDensity.adaptivePlatformDensity,
       fontFamily: 'Roboto', // Custom font
     ),
     home: MyHomePage(),
   );
 }
}

class MyHomePage extends StatefulWidget {
 @override
 _MyHomePageState createState() => _MyHomePageState();
}

class _MyHomePageState extends State<MyHomePage> {
 @override
 Widget build(BuildContext context) {
   return Scaffold(
     appBar: AppBar(
       title: Text('ERP Dashboard'),
     ),
     body: Padding(
       padding: const EdgeInsets.all(16.0),
       child: Column(
         crossAxisAlignment: CrossAxisAlignment.start,
         children: [
           Row(
             children: [
               Text(
                 'Sales Overview',
                 style: TextStyle(fontSize: 24, fontWeight: FontWeight.bold),
               ),
               SizedBox(width: 8),
               Icon(
                 Icons.analytics,
                 size: 40,
                 color: Colors.blue,
               ),
             ],
           ),
           SizedBox(height: 16),
           Row(
             mainAxisAlignment: MainAxisAlignment.spaceBetween,
```

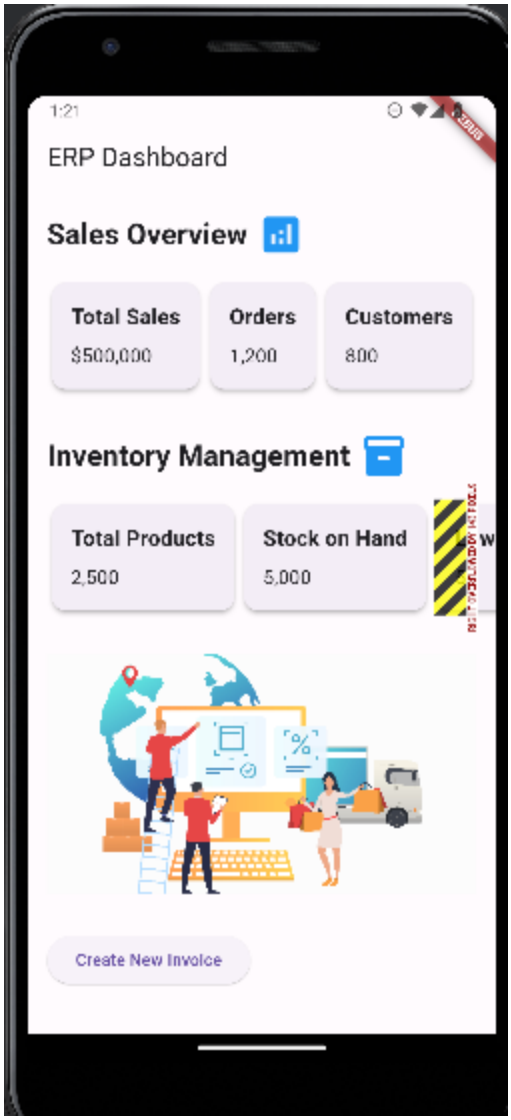
```
            children: [
              _buildStatCard('Total Sales', '\$500,000'),
              _buildStatCard('Orders', '1,200'),
              _buildStatCard('Customers', '800'),
            ],
          ),
          SizedBox(height: 32),
          Row( // Row for "Inventory Management" text and icon
            children: [
              Text(
                'Inventory Management',
                style: TextStyle(fontSize: 24, fontWeight: FontWeight.bold),
              ),
              SizedBox(width: 8),
              Icon(
                Icons.inventory_2_rounded, // inventory_2_rounded icon
                size: 40,
                color: Colors.blue,
              ),
            ],
          ),
          SizedBox(height: 16),
          Row(
            mainAxisAlignment: MainAxisAlignment.spaceBetween,
            children: [
              _buildStatCard('Total Products', '2,500'),
              _buildStatCard('Stock on Hand', '5,000'),
              _buildStatCard('Low Stock Alerts', '50'),
            ],
          ),
          SizedBox(height: 32),
          Image.asset(
            'assets/19238.jpg',
            width: 360,
            height: 200,
            fit: BoxFit.cover,
          ),
          SizedBox(height: 32),
          ElevatedButton(
            onPressed: () {
              Navigator.push(
                context,
                MaterialPageRoute(builder: (context) => InvoiceForm()),
              );
            },
            child: Text('Create New Invoice'),
          ),
        ],
      ),
```

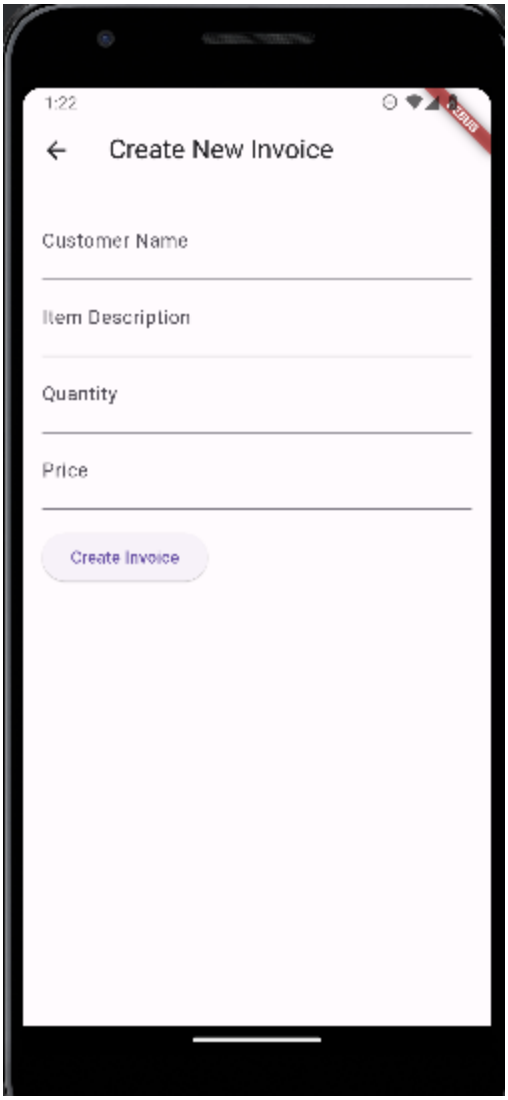
```
    ),
  );
 }

 Widget _buildStatCard(String title, String value) {
   return Card(
     elevation: 3,
     child: Padding(
       padding: const EdgeInsets.all(16.0),
       child: Column(
         crossAxisAlignment: CrossAxisAlignment.start,
         children: [
           Text(
             title,
             style: TextStyle(fontSize: 18, fontWeight: FontWeight.bold),
           ),
           SizedBox(height: 8),
           Text(
             value,
             style: TextStyle(fontSize: 16),
           ),
         ],
       ),
     ),
   );
 }
}

class InvoiceForm extends StatefulWidget {
 @override
 _InvoiceFormState createState() => _InvoiceFormState();
}

class _InvoiceFormState extends State<InvoiceForm> {
 final _formKey = GlobalKey<FormState>();
 TextEditingController _customerNameController = TextEditingController();
 TextEditingController _itemDescriptionController = TextEditingController();
 TextEditingController _quantityController = TextEditingController();
 TextEditingController _priceController = TextEditingController();

 @override
 Widget build(BuildContext context) {
   return Scaffold(
     appBar: AppBar(
       title: Text('Create New Invoice'),
     ),
     body: Padding(
       padding: const EdgeInsets.all(16.0),
       child: Form(
```

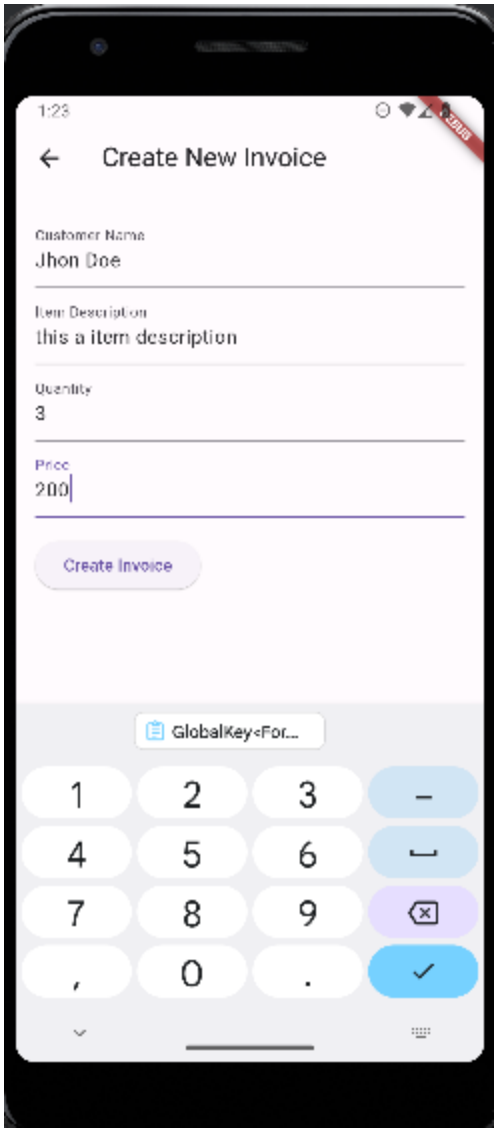
```
        key: _formKey,
        child: Column(
          crossAxisAlignment: CrossAxisAlignment.start,
          children: [
            TextFormField(
              controller: _customerNameController,
              decoration: InputDecoration(labelText: 'Customer Name'),
              validator: (value) {
                if (value?.isEmpty ?? true) {
                  return 'Please enter customer name';
                }
                return null;
              },
            ),
            TextFormField(
              controller: _itemDescriptionController,
              decoration: InputDecoration(labelText: 'Item Description'),
              validator: (value) {
                if (value?.isEmpty ?? true) {
                  return 'Please enter item description';
                }
                return null;
              },
            ),
            TextFormField(
              controller: _quantityController,
              decoration: InputDecoration(labelText: 'Quantity'),
              keyboardType: TextInputType.number,
              validator: (value) {
                if (value?.isEmpty ?? true) {
                  return 'Please enter quantity';
                }
                return null;
              },
            ),
            TextFormField(
              controller: _priceController,
              decoration: InputDecoration(labelText: 'Price'),
              keyboardType: TextInputType.number,
              validator: (value) {
                if (value?.isEmpty ?? true) {
                  return 'Please enter price';
                }
                return null;
              },
            ),
            SizedBox(height: 16),
            ElevatedButton(
              onPressed: () {
```

```
                if (_formKey.currentState!.validate()) {
                  // Form is valid, process the data
                  // Here, you can access the form field values using
                  // _customerNameController.text,
_itemDescriptionController.text,
                  // _quantityController.text, _priceController.text, etc.
                  // Perform your desired actions (e.g., send data to backend)
                  // and navigate to the next screen or show a success
message.
                }
              },
              child: Text('Create Invoice'),
            ),
          ],
        ),
      ),
    ),
  );
 }

 @override
 void dispose() {
   _customerNameController.dispose();
   _itemDescriptionController.dispose();
   _quantityController.dispose();
   _priceController.dispose();
   super.dispose();
 }
}
```

OUTPUT:


1:21
ERP Dashboard
Sales Overview
Total Sales
$500,000
Orders
1,200
Customers
800
Inventory Management
Total Products
2,500
Stock on Hand
5,000
Create New Invoice


1:22
Create New Invoice
Customer Name
Item Description
Quantity
Price
Create Invoice



Conclusion:

Through experimentation with form components in Flutter I understood that Flutter offers a robust set of widgets for building interactive forms efficiently. Widgets such as TextFormField provide convenient ways to capture user input, offering features like validation and input formatting. By wrapping form fields within a Form widget and assigning a GlobalKey<FormState>, it becomes possible to manage form state, validate user input, and trigger actions based on form submission. Additionally, Flutter's flexible layout system allows for seamless integration of form components within the overall app interface. Overall, Flutter's form components empower developers to create intuitive and user-friendly forms tailored to their application's needs with ease.