Bharatiya Vidya Bhavan's
SARDAR PATEL INSTITUTE OF TECHNOLOGY
(Autonomous Institute Affiliated to University of Mumbai)
Munshi Nagar, Andheri (W), Mumbai – 400 058.
Department of Computer Science and Engineering

Experiment	1
Aim	Create Dynamic Registration form
Objective	To Create Dynamic Registration form using UI components and Scaffold elements.
Name	Vedant Ekanath Onkar
UCID	2024510036
Class	FYMCA
Batch	2024-26
Date of Submission	27 March 2025

Technology used	**1. Flutter** **2. VSCode**
Task	Create Dynamic Webpage for submitting review for movie app along with giving ratings for the same. The webpage should include, name, surname, date of birth, address, email id, phone number, gender, review and ratings for the movie in terms of stars/smiley etc. include validation. The information should be fetched and displayed on the next page for viewing it. Use snackbar, alert dialog box, try adding fan button (static) etc.
Code with proper label	

```dart
import 'package:flutter/material.dart';
import 'package:flutter_rating_bar/flutter_rating_bar.dart';
import 'package:intl/intl.dart'; // for date formatting

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

class MyApp extends StatelessWidget {
  // Earthy Warmth Color Palette (#6):
  // Dominant (60%): Pale Peach (#FFE0B2)
  // Secondary (30%): Rich Brown (#5D4037)
  // Accent (10%): Turquoise (#00BCD4)
  final Color dominantColor = Color(0xFFFFE0B2);
  final Color secondaryColor = Color(0xFF5D4037);
  final Color accentColor = Color(0xFF00BCD4);

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Movie Review App',
      theme: ThemeData(
        scaffoldBackgroundColor: dominantColor,
        primaryColor: secondaryColor,
        colorScheme: ColorScheme.light(
          primary: secondaryColor,
          secondary: accentColor,
        ),
        appBarTheme: AppBarTheme(
          backgroundColor: secondaryColor,
          foregroundColor: Colors.white,
        ),
        elevatedButtonTheme: ElevatedButtonThemeData(
          style: ElevatedButton.styleFrom(
            backgroundColor: accentColor,
            foregroundColor: Colors.white,
            padding: EdgeInsets.symmetric(horizontal: 40,
vertical: 12),
            shape: RoundedRectangleBorder(
              borderRadius: BorderRadius.circular(8),
            ),
          ),
        ),
        cardTheme: CardTheme(
          color: Color(0xFFFFF3E0),
          shape: RoundedRectangleBorder(
            borderRadius: BorderRadius.circular(10),
          ),
          elevation: 4,
        ),
        inputDecorationTheme: InputDecorationTheme(
          filled: true,
          fillColor: Colors.white,
          border: OutlineInputBorder(
            borderRadius: BorderRadius.circular(8),
          ),
          contentPadding: EdgeInsets.symmetric(horizontal:
12, vertical: 16),
        ),
      ),
      home: ReviewFormPage(
        dominantColor: dominantColor,
        secondaryColor: secondaryColor,
        accentColor: accentColor,
      ),
    );
  }
}

class ReviewFormPage extends StatefulWidget {
  final Color dominantColor;
  final Color secondaryColor;
  final Color accentColor;

  const ReviewFormPage({
    Key? key,
    required this.dominantColor,
    required this.secondaryColor,
    required this.accentColor,
  }) : super(key: key);

  @override
  _ReviewFormPageState createState() =>
_ReviewFormPageState();
}

class _ReviewFormPageState extends State<ReviewFormPage> {
  final GlobalKey<FormState> _formKey =
GlobalKey<FormState>();

  // Controllers for form fields
  final TextEditingController _firstNameController =
TextEditingController();
  final TextEditingController _surnameController =
TextEditingController();
  final TextEditingController _dobController =
TextEditingController(); // Date of Birth
  final TextEditingController _addressController =
TextEditingController();
  final TextEditingController _emailController =
TextEditingController();
  final TextEditingController _phoneController =
TextEditingController();
  final TextEditingController _reviewController =
TextEditingController();

  // For gender selection
  String? _gender;

  // For movie rating
  double _rating = 5.0;

  // List to store submitted reviews
  final List<Map<String, dynamic>> _reviewRecords = [];

  // Function to pick a date using a date picker
  Future<void> _selectDate(BuildContext context) async {
    DateTime initialDate =
DateTime.now().subtract(Duration(days: 365 * 18));
    final DateTime? picked = await showDatePicker(
      context: context,
      initialDate: initialDate,
      firstDate: DateTime(1900),
      lastDate: DateTime.now(),
    );
    if (picked != null) {
      setState(() {
        _dobController.text =
DateFormat('yyyy-MM-dd').format(picked);
      });
    }
  }

  // Submit the review form
  void _submitReview() {
    if (_formKey.currentState!.validate()) {
      // Save the review record
      Map<String, dynamic> reviewRecord = {
        'firstName': _firstNameController.text,
        'surname': _surnameController.text,
        'dob': _dobController.text,
        'address': _addressController.text,
        'email': _emailController.text,
        'phone': _phoneController.text,
        'gender': _gender,
        'review': _reviewController.text,
        'rating': _rating,
      };

      setState(() {
        _reviewRecords.add(reviewRecord);
      });

      // Show a snackbar confirming submission
      ScaffoldMessenger.of(context).showSnackBar(
        SnackBar(content: Text('Review Submitted
Successfully!')),
      );

      // Show an alert dialog box for confirmation
      showDialog(
        context: context,
        builder: (context) => AlertDialog(
          title: Text('Submission Confirmation'),
          content: Text('Your review has been submitted.'),
          actions: [
            TextButton(
              onPressed: () {
                Navigator.of(context).pop();
                // Navigate to the Review Records page after closing the dialog
                Navigator.push(
                  context,
                  MaterialPageRoute(
                    builder: (context) => ReviewRecordsPage(
                      reviewRecords: _reviewRecords,
                      secondaryColor: widget.secondaryColor,
                      accentColor: widget.accentColor,
                    ),
                  ),
                );
              },
              child: Text('OK'),
            ),
          ],
        ),
      );

      // Optionally, clear the form fields
      _formKey.currentState!.reset();
      _dobController.clear();
      _gender = null;
      setState(() {
        _rating = 5.0;
      });
    }
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Movie Review Submission'),
      ),
      body: SingleChildScrollView(
        padding: EdgeInsets.all(16),
        child: Form(
          key: _formKey,
          child: Column(
            crossAxisAlignment: CrossAxisAlignment.stretch,
            children: [
              Text(
                'Enter Your Details',
                style: TextStyle(
                  fontSize: 20,
                  fontWeight: FontWeight.bold,
                  color: widget.secondaryColor,
                ),
              ),
              SizedBox(height: 10),
              // First Name
              TextFormField(
                controller: _firstNameController,
                decoration: InputDecoration(
                  labelText: 'First Name',
                ),
                validator: (value) =>
                    value == null || value.isEmpty ? 'Please enter your first name' : null,
              ),
              SizedBox(height: 10),
              // Surname
              TextFormField(
                controller: _surnameController,
                decoration: InputDecoration(
                  labelText: 'Surname',
                ),
                validator: (value) =>
                    value == null || value.isEmpty ? 'Please enter your surname' : null,
              ),
              SizedBox(height: 10),
              // Date of Birth with a date picker
              TextFormField(
                controller: _dobController,
                decoration: InputDecoration(
                  labelText: 'Date of Birth (YYYY-MM-DD)',
                  suffixIcon: IconButton(
                    icon: Icon(Icons.calendar_today),
                    onPressed: () => _selectDate(context),
                  ),
                ),
                readOnly: true,
                validator: (value) =>
                    value == null || value.isEmpty ? 'Please select your date of birth' : null,
              ),
              SizedBox(height: 10),
              // Address
              TextFormField(
                controller: _addressController,
                decoration: InputDecoration(
                  labelText: 'Address',
                ),
                maxLines: 2,
                validator: (value) =>
                    value == null || value.isEmpty ? 'Please
enter your address' : null,
              ),
              SizedBox(height: 10),
              // Email ID
              TextFormField(
                controller: _emailController,
                decoration: InputDecoration(
                  labelText: 'Email ID',
                ),
                keyboardType: TextInputType.emailAddress,
                validator: (value) {
                  if (value == null || value.isEmpty) {
                    return 'Please enter your email';
                  }
                  if
(!RegExp(r'\S+@\S+\.\S+').hasMatch(value)) {
                    return 'Please enter a valid email';
                  }
                  return null;
                },
              ),
              SizedBox(height: 10),
              // Phone Number
              TextFormField(
                controller: _phoneController,
                decoration: InputDecoration(
                  labelText: 'Phone Number',
                ),
                keyboardType: TextInputType.phone,
                validator: (value) {
                  if (value == null || value.isEmpty) {
                    return 'Please enter your phone number';
                  }
                  if (!RegExp(r'^\d{10}$').hasMatch(value))
{
```

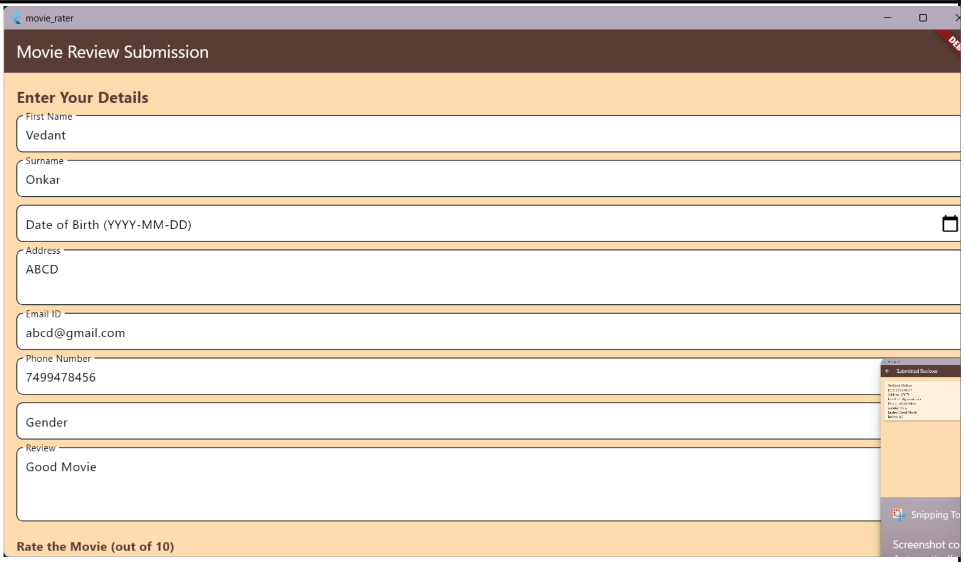
```
                    return 'Please enter a valid 10-digit
phone number';
                  }
                  return null;
                },
              ),
              SizedBox(height: 10),
              // Gender Selection using Dropdown
              DropdownButtonFormField<String>(
                value: _gender,
                decoration: InputDecoration(
                  labelText: 'Gender',
                ),
                items: ['Male', 'Female', 'Other']
                    .map((gender) => DropdownMenuItem(
                          value: gender,
                          child: Text(gender),
                        ))
                    .toList(),
                onChanged: (val) {
                  setState(() {
                    _gender = val;
                  });
                },
                validator: (value) =>
                    value == null || value.isEmpty ? 'Please
select your gender' : null,
              ),
              SizedBox(height: 10),
              // Review (multi-line)
              TextFormField(
                controller: _reviewController,
                decoration: InputDecoration(
                  labelText: 'Review',
                ),
                maxLines: 3,
                validator: (value) =>
                    value == null || value.isEmpty ? 'Please
enter your review' : null,
              ),
              SizedBox(height: 20),
              Text(
                'Rate the Movie (out of 10)',
                style: TextStyle(
                  fontSize: 16,
                  fontWeight: FontWeight.bold,
                  color: widget.secondaryColor,
                ),
              ),
              SizedBox(height: 10),
              // Rating Bar (using stars) with 10 items and
half rating allowed
              RatingBar.builder(
                initialRating: _rating,
                minRating: 0.5,
                direction: Axis.horizontal,
                allowHalfRating: true,
                itemCount: 10,
                itemPadding:
EdgeInsets.symmetric(horizontal: 2.0),
                itemBuilder: (context, _) => Icon(
                  Icons.star,
                  color: widget.accentColor,
                ),
                onRatingUpdate: (rating) {
                  setState(() {
                    _rating = rating;
                  });
                },
              ),
              SizedBox(height: 10),
              // Display current rating value
              Text(
                'Rating Value:
${_rating.toStringAsFixed(1)}',
                style: TextStyle(
                  fontSize: 16,
                  color: widget.secondaryColor,
                ),
              ),
              SizedBox(height: 20),
              ElevatedButton(
                onPressed: _submitReview,
                child: Text('Submit Review'),
              ),
            ],
          ),
        ),
      ),
    );
  }
}
```

```
class ReviewRecordsPage extends StatelessWidget {
  final List<Map<String, dynamic>> reviewRecords;
  final Color secondaryColor;
  final Color accentColor;

  const ReviewRecordsPage({
    Key? key,
    required this.reviewRecords,
    required this.secondaryColor,
    required this.accentColor,
  }) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Submitted Reviews'),
      ),
      body: reviewRecords.isEmpty
          ? Center(
              child: Text(
                'No reviews available.',
                style: TextStyle(
                  fontSize: 18,
                  color: secondaryColor,
                ),
              ),
            )
          : ListView.builder(
              padding: EdgeInsets.all(16),
              itemCount: reviewRecords.length,
              itemBuilder: (context, index) {
                final record = reviewRecords[index];
                return Card(
                  margin: EdgeInsets.only(bottom: 12),
                  child: ListTile(
                    title: Text(
                      '${record['firstName']}
${record['surname']}',
                      style: TextStyle(
                        fontWeight: FontWeight.bold,
                        color: secondaryColor,
                      ),
                    ),
                    subtitle: Column(
                      crossAxisAlignment:
CrossAxisAlignment.start,
                      children: [
                        Text('DOB: ${record['dob']}'),
                        Text('Address:
${record['address']}'),
                        Text('Email: ${record['email']}'),
                        Text('Phone: ${record['phone']}'),
                        Text('Gender: ${record['gender']}'),
                        Text('Review: ${record['review']}'),
                        Text('Rating:
${record['rating'].toStringAsFixed(1)}'),
                      ],
                    ),
                  ),
                );
              },
            ),
    );
  }
}
```

Screenshots



Movie Review Submission

Enter Your Details

First Name: Vedant

Surname: Onkar

Date of Birth (YYYY-MM-DD)

Address: ABCD

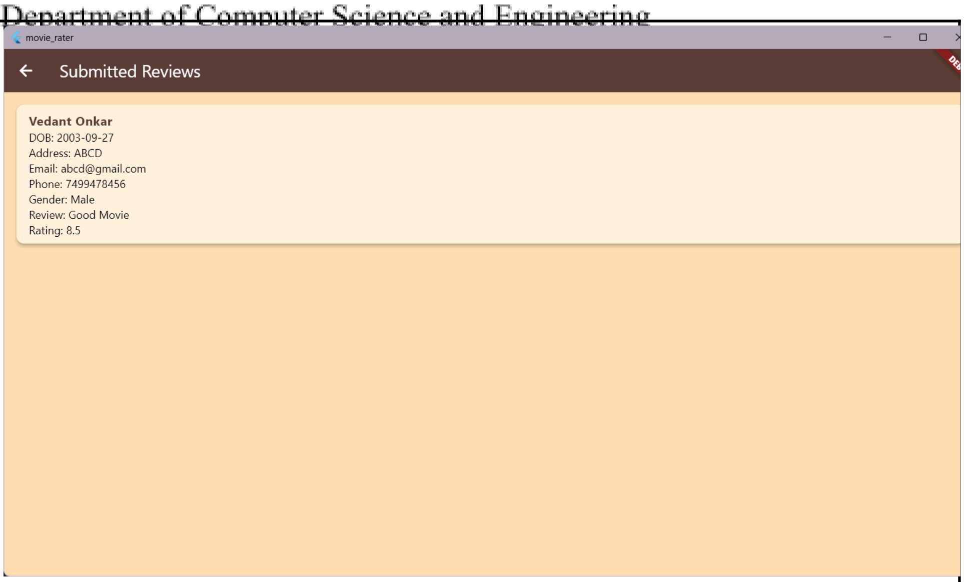
Email ID: abcd@gmail.com

Phone Number: 7499478456

Gender

Review: Good Movie

Rate the Movie (out of 10)

	 Submitted Reviews **Vedant Onkar** DOB: 2003-09-27 Address: ABCD Email: abcd@gmail.com Phone: 7499478456 Gender: Male Review: Good Movie Rating: 8.5
Question and Answers	Answer the following Questions: 1. How to create snackbar? Which component of flutter it belongs to?A snackbar is created using ScaffoldMessenger.of(context).showSnackBar() with a SnackBar widget that contains the message. This component is closely tied to the Scaffold, which provides the layout for the application. 2. How to give validation to any field in flutter? Validation for any field is implemented by adding a validator property to the TextFormField. The validator function checks the input value and returns an error message if the field does not meet the required criteria. 3. How to navigate from one page to another. Navigation between pages is achieved using Navigator.push with a MaterialPageRoute. When the Navigator.push method is called, the app transitions to a new page; similarly, Navigator.pop is used to return to the previous page. 4. How are values of variables fetched in flutter. Values of variables in Flutter are typically fetched from controllers, such as TextEditingController, by accessing the .text property of the controller. Additionally, state variables are maintained and updated using setState, which ensures that any changes in variable values are reflected in the UI.
Conclusion	In conclusion, the provided code illustrates essential Flutter features including user feedback with snackbars, form field validation, page navigation, and dynamic value management. These techniques form the backbone of interactive and responsive Flutter applications.