BVRIT HYDERABAD College of Engineering for Women

Title: 7. a) Design a file with various input fields.　　　　Date:

　　　b) Implement form validation and error handling.　　　Page No.:

Code:

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

void main() => runApp(MyApp());
class MyApp extends StatelessWidget {
 @override
 Widget build(BuildContext context) {
  return MaterialApp(
    title: 'Enhanced Registration Form',
    home: Scaffold(
      appBar: AppBar(title: Text('User Registration')),
      body: Padding(
        padding: EdgeInsets.all(16.0),
        child: UserForm(),
      ), ),);}}

class UserForm extends StatefulWidget {
 @override
 _UserFormState createState() => _UserFormState();
}

class _UserFormState extends State<UserForm> {
 final _formKey = GlobalKey<FormState>();
 String? selectedGender;
 bool acceptedTerms = false;
 DateTime? selectedDate;

 final TextEditingController passwordController = TextEditingController();
 final TextEditingController confirmPasswordController =
    TextEditingController();

Future<void> _submitForm() async {
  try {
   if (_formKey.currentState!.validate()) {
    if (!acceptedTerms) {
     ScaffoldMessenger.of(context).showSnackBar(
       SnackBar(content: Text('Please accept the terms and conditions')),
     );
     return;
    }
    ScaffoldMessenger.of(context).showSnackBar(
     SnackBar(content: Text('Form Submitted Successfully')),
    );
}}
```

```
    catch (e) {
ScaffoldMessenger.of(context).showSnackBar(
  SnackBar(content: Text('An error occurred: $e')),
);} }
     Future<void> _pickDate() async {
       final DateTime? picked = await showDatePicker(
         context: context,
         initialDate: DateTime(2000),
         firstDate: DateTime(1900),
         lastDate: DateTime.now(),
       );
       if (picked != null) {
       setState(() {
         selectedDate = picked;
       });} }
    @override
     Widget build(BuildContext context) {
      return Form(
       key: _formKey,
       child: ListView(
        children: [
         TextFormField(
           decoration: InputDecoration(labelText: 'Username'),
           validator: (value) => value == null || value.trim().isEmpty
             ? 'Username is required'
             : null,
         ),
         TextFormField(
           decoration: InputDecoration(labelText: 'First Name'),
           validator: (value) => value == null || value.trim().isEmpty
             ? 'First name is required'
             : null,
         ),
         TextFormField(
           decoration: InputDecoration(labelText: 'Last Name'),
           validator: (value) => value == null || value.trim().isEmpty
             ? 'Last name is required'
             : null,
         ),
         TextFormField(
           decoration: InputDecoration(labelText: 'Contact Number'),
           keyboardType: TextInputType.phone,
           validator: (value) {
            if (value == null || value.trim().isEmpty)
              return 'Contact number is required';
```

```
    if (!RegExp(r'^\d{10}$').hasMatch(value))
      return 'Enter a valid 10-digit number';
    return null;
  }, ),
  TextFormField(
    decoration: InputDecoration(labelText: 'Email'),
    keyboardType: TextInputType.emailAddress,
    validator: (value) {
     if (value == null || value.trim().isEmpty)
       return 'Email is required';
     if (!RegExp(r'^[^@]+@[^@]+\.[^@]+').hasMatch(value))
       return 'Enter a valid email';
     return null;
  }, ),
  TextFormField(
    decoration: InputDecoration(labelText: 'Age'),
    keyboardType: TextInputType.number,
    validator: (value) {
     if (value == null || value.trim().isEmpty)
       return 'Age is required';
     int? age = int.tryParse(value);
     if (age == null || age < 1 || age > 120)
       return 'Enter a valid age';
     return null;
  }, ),
  TextFormField(
    controller: passwordController,
    decoration: InputDecoration(labelText: 'Password'),
    obscureText: true,
    validator: (value) {
     if (value == null || value.isEmpty) return 'Password is required';
     if (value.length < 6)
       return 'Password must be at least 6 characters';
     return null;
  }, ),
  TextFormField(
    controller: confirmPasswordController,
    decoration: InputDecoration(labelText: 'Confirm Password'),
    obscureText: true,
    validator: (value) {
     if (value != passwordController.text)
       return 'Passwords do not match';
     return null;
  }, ),
  TextFormField(
```

```dart
   decoration: InputDecoration(labelText: 'Address'),
  maxLines: 2,
  validator: (value) => value == null || value.trim().isEmpty
    ? 'Address is required'
    : null,
 ),
 TextFormField(
  readOnly: true,
  decoration: InputDecoration(
   labelText: 'Date of Birth',
   suffixIcon: Icon(Icons.calendar_today),
  ),
  controller: TextEditingController(
   text: selectedDate == null
     ? ''
     : '${selectedDate!.day}/${selectedDate!.month}/${selectedDate!.year}',
  ),
  onTap: () async {
   final DateTime? picked = await showDatePicker(
    context: context,
    initialDate: DateTime(2000),
    firstDate: DateTime(1900),
    lastDate: DateTime.now(),
   );
   if (picked != null) {
    setState(() {
     selectedDate = picked;
    });} },
  validator: (value) {
   if (selectedDate == null) {
    return 'Please select your date of birth';
   }
   return null;
  },),
 DropdownButtonFormField<String>(
  decoration: InputDecoration(labelText: 'Gender'),
  items: ['Male', 'Female', 'Other']
    .map((gender) =>
      DropdownMenuItem(value: gender, child: Text(gender)))
    .toList(),
  onChanged: (value) => setState(() => selectedGender = value),
  validator: (value) =>
    value == null ? 'Please select a gender' : null,
 ),
```

```
CheckboxListTile(
    title: Text('I accept the Terms & Conditions'),
    value: acceptedTerms,
    onChanged: (value) =>
        setState(() => acceptedTerms = value ?? false),
   ),
   SizedBox(height: 20),
   ElevatedButton(
     onPressed: _submitForm,
     child: Text('Register'),
   ),
  ],
 ),
);}
```

Output:

BVRIT HYDERABAD College of Engineering for Women

Title: 8. a) Add animations to UI elements using Flutter animation framework.   Date:

Page No.:

Code:

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

void main() => runApp(const MainApp());

class MainApp extends StatefulWidget {
  const MainApp({super.key});

  @override
  State<MainApp> createState() => _MainAppState();
}

class _MainAppState extends State<MainApp> {
  bool _isBig = false;

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      home: Scaffold(
        appBar: AppBar(
          title: const Text('Beginner Animation'),
        ),
        body: Center(
          child: Column(
            mainAxisAlignment: MainAxisAlignment.center,
            children: [
              const Text(
                'Tap the circle to animate',
                style: TextStyle(fontSize: 18),
              ),
              const SizedBox(height: 20),
              GestureDetector(
                onTap: () {
                  setState(() {
                    _isBig = !_isBig;
                  }); },
                child: AnimatedContainer(
                  duration: const Duration(seconds: 1),
                  curve: Curves.easeInOutBack,
                  width: _isBig ? 250 : 100,
                  height: _isBig ? 250 : 100,
                  decoration: BoxDecoration(
```

Title:

Date:
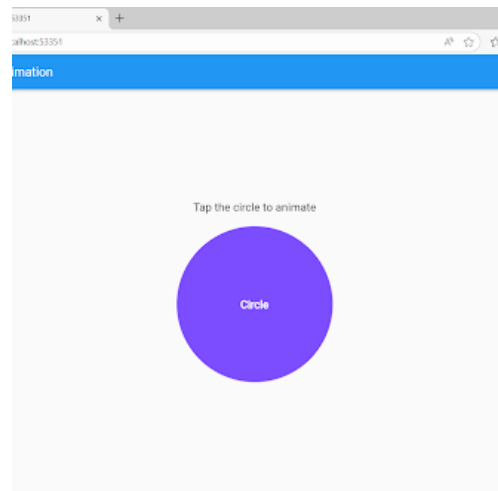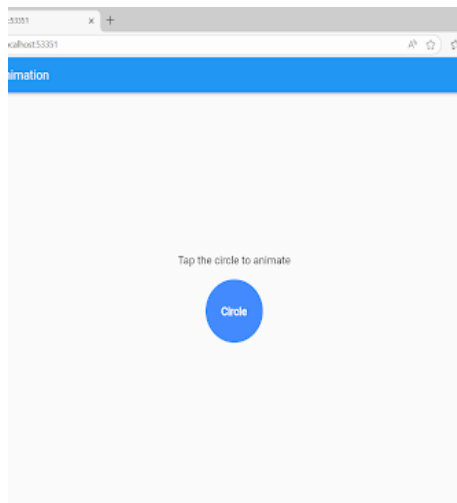
Page No.:

```
shape: BoxShape.circle,
      color: _isBig
        ? Colors.deepPurpleAccent
        : Colors.blueAccent,
    ),
    alignment: Alignment.center,
    child: const Text(
     'Circle',
     style: TextStyle(
       color: Colors.white,
       fontWeight: FontWeight.bold,
       fontSize: 18,
      ),
     ),
    ),
   ),
  ],
 ),),),),); }}
```

Output:

Code:

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

void main() {
  runApp(const MyApp());
}
class MyApp extends StatelessWidget {
  const MyApp({super.key});
 @override
  Widget build(BuildContext context) {
    return const MaterialApp(
      home: AnimationDemo(),
    );}}
class AnimationDemo extends StatefulWidget {
  const AnimationDemo({super.key});
@override
  State<AnimationDemo> createState() => _AnimationDemoState();
}
class _AnimationDemoState extends State<AnimationDemo>
    with TickerProviderStateMixin {
  late AnimationController _scaleController;
  late Animation<double> _scaleAnimation;
  late AnimationController _fadeController;
  late Animation<double> _fadeAnimation;
  late AnimationController _slideController;
  late Animation<Offset> _slideAnimation;
  late AnimationController _rotateController;
  late Animation<double> _rotateAnimation;
  bool _visible = false;
  bool _sliding = false;
  bool _rotating = false;
  bool _scaling = true;
@override
  void initState() {
    super.initState();
    _fadeController = AnimationController(
      vsync: this,
      duration: const Duration(seconds: 1),
    );
    _fadeAnimation = Tween<double>(begin: 1, end: 0).animate(_fadeController);
    _slideController = AnimationController(
      vsync: this,
      duration: const Duration(seconds: 1),
    );
```

```dart
    _slideAnimation = Tween<Offset>(
      begin: const Offset(-1.0, 0.0),
      end: const Offset(1.0, 0.0),
    ).animate(_slideController);
    _rotateController = AnimationController(
      vsync: this,
      duration: const Duration(seconds: 2),
    );
    _rotateAnimation =
      Tween<double>(begin: 0, end: 1).animate(_rotateController);
    _scaleController = AnimationController(
      vsync: this,
      duration: const Duration(seconds: 2),
    )..repeat(reverse: true);
    _scaleAnimation = CurvedAnimation(
      parent: _scaleController,
      curve: Curves.easeInOut,
    );}
  @override
  void dispose() {
    _scaleController.dispose();
    _fadeController.dispose();
    _slideController.dispose();
    _rotateController.dispose();
    super.dispose();
  }
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: const Text('Animation Demo'),
      ),
      body: Center(
        child: Column(
          mainAxisAlignment: MainAxisAlignment.center,
          children: [
            FadeTransition(
              opacity: _fadeAnimation,
              child: SlideTransition(
                position: _slideAnimation,
                child: RotationTransition(
                  turns: _rotateAnimation,
                  child: ScaleTransition(
                    scale: _scaleAnimation,
                    child: Image.network(
```

```
    'https://upload.wikimedia.org/wikipedia/commons/2/2f/Google_2015_logo.svg',
        width: 100,
      ),),), ),
  SwitchListTile(
   title: const Text('Fade'),
   value: _visible,
   onChanged: (bool value) {
    setState(() {
      _visible = value;
      _visible
        ? _fadeController.forward()
        : _fadeController.reverse();
    });},),
  SwitchListTile(
   title: const Text('Slide'),
   value: _sliding,
   onChanged: (bool value) {
    setState(() {
      _sliding = value;
      _sliding
        ? _slideController.repeat(reverse: true)
        : _slideController.stop();
    });}, ),
  SwitchListTile(
   title: const Text('Rotate'),
   value: _rotating,
   onChanged: (bool value) {
    setState(() {
      _rotating = value;
      _rotating
        ? _rotateController.repeat()
        : _rotateController.stop();
    });},),
  SwitchListTile(
   title: const Text('Scale'),
   value: _scaling,
   onChanged: (bool value) {
    setState(() {
      _scaling = value;
      _scaling
        ? _scaleController.repeat(reverse: true)
        : _scaleController.stop();
    });
   }, ), ), ] ) ) ) }
```

Title: Date:

                                              Page No.:

Output:

BVRIT HYDERABAD College of Engineering for Women

Code:

```dart
import 'dart:convert';
import 'package:http/http.dart' as http;

void main() async {
  // API endpoint for users
  final url = Uri.https(
    'jsonplaceholder.typicode.com',
    '/users',
  );
  try {
    final response = await http.get(url);
    if (response.statusCode == 200) {
      final List<dynamic> jsonData = jsonDecode(response.body);
      for (var user in jsonData) {
        final id = user['id'];
        final name = user['name'];
        final phone = user['phone'];
        print('ID: $id, Name: $name, Contact: $phone');
      }
    } else {
      print('Request failed with code: ${response.statusCode}.');
    }
  } catch (e) {
    print('Error: $e');
  }
}
```

Output:

```
ID: 1, Name: Leanne Graham, Contact: 1-770-736-8031 x56442
ID: 2, Name: Ervin Howell, Contact: 010-692-6593 x09125
ID: 3, Name: Clementine Bauch, Contact: 1-463-123-4447
ID: 4, Name: Patricia Lebsack, Contact: 493-170-9623 x156
ID: 5, Name: Chelsey Dietrich, Contact: (254)954-1289
ID: 6, Name: Mrs. Dennis Schulist, Contact: 1-477-935-8478 x6430
ID: 7, Name: Kurtis Weissnat, Contact: 210.067.6132
ID: 8, Name: Nicholas Runolfsdottir V, Contact: 586.493.6943 x140
ID: 9, Name: Glenna Reichert, Contact: (775)976-6794 x41206
ID: 10, Name: Clementina DuBuque, Contact: 024-648-3804
```

BVRIT HYDERABAD College of Engineering for Women

Title: 9.b) Display the fetched data in a meaningful way in the UI.

Date:

Page No.:

Code:

```dart
import 'dart:convert' as convert;
import 'package:http/http.dart' as http;

void main() async {

  final url = Uri.https(
    'jsonplaceholder.typicode.com',
    '/users',
  );

  final response = await http.get(url);

  if (response.statusCode == 200) {
    // Decode JSON response
    final List users = convert.jsonDecode(response.body);

    print('===== Users List =====\n');

    // Loop through users and print details
    for (var user in users) {
      final id = user['id'];
      final name = user['name'];
      final phone = user['phone'];

        print('ID: $id');
      print('Name: $name');
      print('Contact: $phone');
      print('-------------------');
    }
  } else {
    print('Request failed with code: ${response.statusCode}.');
  }
}}}
```

Output:

```
===== Users List =====

ID: 1
Name: Leanne Graham
Contact: 1-770-736-8031 x56442
--------------------
ID: 2
Name: Ervin Howell
Contact: 010-692-6593 x09125
--------------------
ID: 3
Name: Clementine Bauch
Contact: 1-463-123-4447
--------------------
ID: 4
Name: Patricia Lebsack
Contact: 493-170-9623 x156
--------------------
ID: 5
Name: Chelsey Dietrich
Contact: (254)954-1289
--------------------
ID: 6
Name: Mrs. Dennis Schulist
Contact: 1-477-935-8478 x6430
--------------------
ID: 7
Name: Kurtis Weissnat
Contact: 210.067.6132
--------------------
```

```
ID: 8
Name: Nicholas Runolfsdottir V
Contact: 586.493.6943 x140
--------------------
ID: 9
Name: Glenna Reichert
Contact: (775)976-6794 x41206
--------------------
ID: 10
Name: Clementina DuBuque
Contact: 024-648-3804
--------------------
```