



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	5
Aim	Create an app for MCA department using side and tabs navigation
Objective	<ul style="list-style-type: none">• To Create MCA App using UI Component• To Develop SideDrawer and Tabs Navigation in one application• To Use 60-30-10 rule• To Use Webview concept• To Use Expandable Card View Concept
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 an app for MCA department using side and tabs navigation consisting of the following: <ol style="list-style-type: none">1) Faculty information with Expandable Card view2) Department Vision and Mission with expandable Card view3) Notification of any upcoming event (links to the website).4) WebView5) Feedback form along with rating
Code with proper label	<pre>import 'package:flutter/material.dart'; import 'package:file_picker/file_picker.dart'; // A simple model to hold registration data. class RegistrationData { String name; String email; String contact; String rollNo; String hscCollege; String hscYear; double hscTotal; double hscOutOf; double hscPercentage; String sscCollege; String sscYear; double sscTotal; double sscOutOf; double sscPercentage; // Graduation details bool isGraduationCGPA; // true if using CGPA, false for</pre>



percentage.

```
List<double> gradList; // marks for 1-8 semesters.  
double gradAggregate;  
String resumeFile;  
String additionalCourses;
```

```
RegistrationData({  
    required this.name,  
    required this.email,  
    required this.contact,  
    required this.rollNo,  
    required this.hscCollege,  
    required this.hscYear,  
    required this.hscTotal,  
    required this.hscOutOf,  
    required this.hscPercentage,  
    required this.sscCollege,  
    required this.sscYear,  
    required this.sscTotal,  
    required this.sscOutOf,  
    required this.sscPercentage,  
    required this.isGraduationCGPA,  
    required this.gradList,  
    required this.gradAggregate,  
    required this.resumeFile,  
    required this.additionalCourses,  
});  
}
```

```
// Global variable to store registration data.  
RegistrationData? registrationData;
```

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

```
class TPOApp extends StatelessWidget {  
    @override  
    Widget build(BuildContext context) {  
        return MaterialApp(  
            title: 'TPO App',  
            theme: ThemeData(  
                primarySwatch: Colors.indigo,  
                colorScheme: ColorScheme.fromSwatch(primarySwatch:  
Colors.indigo)  
                .copyWith(secondary: Colors.orangeAccent),  
            ),  
        );  
    }  
}
```



```
visualDensity:
VisualDensity.adaptivePlatformDensity,
),
// Set the HomeScreen as the initial route
home: HomeScreen(),
routes: {
  '/registration': (context) => RegistrationPage(),
  '/companyDetails': (context) =>
CompanyDetailsPage(),
  '/myDetails': (context) => MyDetailsPage(),
  '/resumeView': (context) => ResumeViewPage(),
},
);
}
}

// New HomeScreen Widget with a gradient background.
class HomeScreen extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: Container(
        decoration: BoxDecoration(
          gradient: LinearGradient(
            colors: [Colors.indigo.shade200,
Colors.orangeAccent.shade100],
            begin: Alignment.topLeft,
            end: Alignment.bottomRight,
          ),
        ),
      child: Center(
        child: Padding(
          padding: const EdgeInsets.all(16),
          child: Column(
            mainAxisAlignment: MainAxisAlignment.center,
            children: [
              Text(
                'Welcome to TPO App',
                style: TextStyle(
                  fontSize: 32,
                  fontWeight: FontWeight.bold,
                  color: Colors.white,
                ),
                textAlign: TextAlign.center,
              ),
              SizedBox(height: 20),
            ],
          ),
        ),
      ),
    );
  }
}
```



```
Text(
  'Manage your registrations, view company
  details, and track your academic information easily.',
  textAlign: TextAlign.center,
  style: TextStyle(fontSize: 18, color:
Colors.white70),
),
  SizedBox(height: 40),
  ElevatedButton(
    style: ElevatedButton.styleFrom(
      backgroundColor: Colors.orangeAccent,
      padding:
EdgeInsets.symmetric(horizontal: 32, vertical: 16),
      textStyle: TextStyle(fontSize: 18),
    ),
    onPressed: () {
      // Replace the HomeScreen with
      HomePage when Get Started is pressed
      Navigator.pushReplacement(
        context,
        MaterialPageRoute(builder: (_) =>
HomePage()),
      );
    },
    child: Text('Get Started'),
  ),
],
),
),
),
),
),
);
}
}

// HomePage with a side navigation drawer
class HomePage extends StatefulWidget {
  @override
  _HomePageState createState() => _HomePageState();
}

class _HomePageState extends State<HomePage> {
  int _selectedIndex = 0;

  final _pages = [
    RegistrationPage(),
```



```
CompanyDetailsPage(),
MyDetailsPage(),
];

void _onItemTapped(int index) {
  setState(() {
    _selectedIndex = index;
  });
}

// Drawer Widget to navigate between pages
Widget _buildDrawer() {
  return Drawer(
    child: ListView(
      padding: EdgeInsets.zero,
      children: <Widget>[
        DrawerHeader(
          decoration: BoxDecoration(
            gradient: LinearGradient(
              colors: [Colors.indigo,
Colors.orangeAccent],
              begin: Alignment.topLeft,
              end: Alignment.bottomRight,
            ),
          ),
        child: Text(
          'TPO App Menu',
          style: TextStyle(
            color: Colors.white,
            fontSize: 26,
          ),
        ),
        ListTile(
          leading: Icon(Icons.app_registration, color:
Colors.indigo),
          title: Text('Registration'),
          onTap: () {
            Navigator.pop(context);
            _onItemTapped(0);
          },
        ),
        ListTile(
          leading: Icon(Icons.business, color:
Colors.indigo),
          title: Text('Company Details'),
```



```
onTap: () {
    Navigator.pop(context);
    _onItemTapped(1);
  },
),
ListTile(
  leading: Icon(Icons.person, color:
Colors.indigo),
  title: Text('My Details'),
  onTap: () {
    Navigator.pop(context);
    _onItemTapped(2);
  },
),
],
),
);
}

@override
Widget build(BuildContext context) {
  return Scaffold(
    drawer: _buildDrawer(),
    appBar: AppBar(
      title: Text('TPO App'),
      backgroundColor: Colors.indigo,
    ),
    body: _pages[_selectedIndex],
  );
}

// Registration Page with form fields and enhanced UI.
class RegistrationPage extends StatefulWidget {
  @override
  _RegistrationPageState createState() =>
  _RegistrationPageState();
}

class _RegistrationPageState extends
State<RegistrationPage> {
  final _formKey = GlobalKey<FormState>();

  // Controllers for HSC and SSC to enable live percentage
  calculation.
  final TextEditingController _hscTotalController =
```



```
TextEditingController();  
    final TextEditingController _hscOutOfController =  
TextEditingController();  
    final TextEditingController _sscTotalController =  
TextEditingController();  
    final TextEditingController _sscOutOfController =  
TextEditingController();  
  
    // Form fields  
String name = '';  
String email = '';  
String contact = '';  
String rollNo = '';  
  
    // HSC details  
String hscCollege = '';  
String hscYear = '';  
double hscTotal = 0;  
double hscOutOf = 0;  
double hscPercentage = 0;  
  
    // SSC details  
String sscCollege = '';  
String sscYear = '';  
double sscTotal = 0;  
double sscOutOf = 0;  
double sscPercentage = 0;  
  
    // Graduation details: option to select CGPA or  
Percentage.  
    bool isGraduationCGPA = true; // true: CGPA, false:  
Percentage.  
    // Assume 8 semesters for graduation.  
List<double> gradList = List.filled(8, 0.0);  
double gradAggregate = 0;  
  
    // Resume file name  
String resumeFile = '';  
  
    // Additional courses  
String additionalCourses = '';  
  
@override  
void initState() {  
    super.initState();  
    // Add listeners to auto-calc HSC and SSC percentages.
```



```
_hscTotalController.addListener(_updateHscPercentage);
_hscOutOfController.addListener(_updateHscPercentage);
_sscTotalController.addListener(_updateSscPercentage);
_sscOutOfController.addListener(_updateSscPercentage);
}

@Override
void dispose() {
    _hscTotalController.dispose();
    _hscOutOfController.dispose();
    _sscTotalController.dispose();
    _sscOutOfController.dispose();
    super.dispose();
}

// Calculate percentage helper
double calculatePercentage(double total, double outOf) {
    if (outOf != 0) {
        return (total / outOf) * 100;
    }
    return 0;
}

// Update HSC percentage based on text field values.
void _updateHscPercentage() {
    final total =
double.tryParse(_hscTotalController.text) ?? 0;
    final outOf =
double.tryParse(_hscOutOfController.text) ?? 0;
    setState(() {
        hscPercentage = calculatePercentage(total, outOf);
    });
}

// Update SSC percentage based on text field values.
void _updateSscPercentage() {
    final total =
double.tryParse(_sscTotalController.text) ?? 0;
    final outOf =
double.tryParse(_sscOutOfController.text) ?? 0;
    setState(() {
        sscPercentage = calculatePercentage(total, outOf);
    });
}

// Calculate graduation aggregate (average over all
```




semesters)

```
void _calculateGraduationAggregate() {
    double sum = 0;
    int count = 0;
    for (var mark in gradList) {
        sum += mark;
        count++;
    }
    gradAggregate = count > 0 ? sum / count : 0;
}

// Function to pick a file for resume
Future<void> _pickResume() async {
    FilePickerResult? result = await
FilePicker.platform.pickFiles(
    type: FileType.custom,
    allowedExtensions: ['pdf', 'doc', 'docx'],
);
    if (result != null &&
result.files.single.name.isNotEmpty) {
        setState(() {
            resumeFile = result.files.single.name;
        });
    }
}

void _submitForm() {
    // Save form fields.
    _formKey.currentState!.save();

    // Update HSC/SSC percentages in case user didn't wait
for auto-update.
    hscTotal = double.tryParse(_hscTotalController.text)
?? 0;
    hscOutOf = double.tryParse(_hscOutOfController.text)
?? 0;
    hscPercentage = calculatePercentage(hscTotal,
hscOutOf);

    sscTotal = double.tryParse(_sscTotalController.text)
?? 0;
    sscOutOf = double.tryParse(_sscOutOfController.text)
?? 0;
    sscPercentage = calculatePercentage(sscTotal,
sscOutOf);
```



```
// Calculate aggregate for graduation
_calculateGraduationAggregate();

// Save entered data to the global registrationData
variable.
registrationData = RegistrationData(
  name: name,
  email: email,
  contact: contact,
  rollNo: rollNo,
  hscCollege: hscCollege,
  hscYear: hscYear,
  hscTotal: hscTotal,
  hscOutOf: hscOutOf,
  hscPercentage: hscPercentage,
  sscCollege: sscCollege,
  sscYear: sscYear,
  sscTotal: sscTotal,
  sscOutOf: sscOutOf,
  sscPercentage: sscPercentage,
  isGraduationCGPA: isGraduationCGPA,
  gradList: gradList,
  gradAggregate: gradAggregate,
  resumeFile: resumeFile,
  additionalCourses: additionalCourses,
);

// Show a dialog confirming submission.
showDialog(
  context: context,
  builder: (_) => AlertDialog(
    backgroundColor: Colors.orangeAccent.shade100,
    title: Text('Registration Successful',
      style: TextStyle(color: Colors.indigo)),
    content: Text(
      'Name: $name\nEmail: $email\nRoll No:
$rollNo\nGraduation Aggregate:
${gradAggregate.toStringAsFixed(2)}',
      style: TextStyle(color: Colors.indigo.shade700),
    ),
    actions: [
      TextButton(
        child: Text('OK', style: TextStyle(color:
Colors.indigo)),
        onPressed: () {
          Navigator.of(context).pop();
        }
      )
    ]
  )
);
```



```
    },  
  ),  
],  
),  
);  
}  
  
Widget _buildNumberField({  
  required String label,  
  required Function(String?) onSave,  
  String? initialValue,  
  String? hintText,  
  TextEditingController? controller,  
}) {  
  return TextFormField(  
    controller: controller,  
    initialValue: controller == null ? initialValue :  
null,  
    keyboardType: TextInputType.number,  
    decoration: InputDecoration(  
      labelText: label,  
      hintText: hintText,  
      border: OutlineInputBorder(),  
    ),  
    onSave: onSave,  
  );  
}  
  
@override  
Widget build(BuildContext context) {  
  final textStyle =  
Theme.of(context).textTheme.titleLarge?.copyWith(color:  
Colors.indigo);  
  
  return SingleChildScrollView(  
    padding: EdgeInsets.all(16),  
    child: Card(  
      elevation: 6,  
      color: Colors.white,  
      shape: RoundedRectangleBorder(borderRadius:  
BorderRadius.circular(16)),  
      child: Padding(  
        padding: EdgeInsets.all(20),  
        child: Form(  
          key: _formKey,
```



```
child: Column(  
  crossAxisAlignment:  
CrossAxisAlignment.stretch,  
  children: [  
    Text('Personal Information', style:  
textStyle),  
    SizedBox(height: 10),  
    TextFormField(  
      decoration: InputDecoration(  
        labelText: 'Name',  
        hintText: 'Enter your full name',  
        border: OutlineInputBorder(),  
      ),  
      onSave: (value) => name = value ?? '',  
    ),  
    SizedBox(height: 10),  
    TextFormField(  
      decoration: InputDecoration(  
        labelText: 'E-mail',  
        hintText: 'example@mail.com',  
        border: OutlineInputBorder(),  
      ),  
      onSave: (value) => email = value ?? '',  
    ),  
    SizedBox(height: 10),  
    TextFormField(  
      decoration: InputDecoration(  
        labelText: 'Contact',  
        hintText: 'Enter your contact number',  
        border: OutlineInputBorder(),  
      ),  
      onSave: (value) => contact = value ??  
'',  
    ),  
    SizedBox(height: 10),  
    TextFormField(  
      decoration: InputDecoration(  
        labelText: 'Roll No',  
        hintText: 'Enter your roll number',  
        border: OutlineInputBorder(),  
      ),  
      onSave: (value) => rollNo = value ??  
'',  
    ),  
    Divider(height: 30, thickness: 2, color:  
Colors.orangeAccent),
```



```
// HSC details
Text('HSC Details', style: textStyle),
SizedBox(height: 10),
TextFormField(
  decoration: InputDecoration(
    labelText: 'College Name',
    hintText: 'Enter HSC college name',
    border: OutlineInputBorder(),
  ),
  onSave: (value) => hscCollege = value
?? '',
),
SizedBox(height: 10),
TextFormField(
  decoration: InputDecoration(
    labelText: 'Year of Passing',
    hintText: 'YYYY',
    border: OutlineInputBorder(),
  ),
  onSave: (value) => hscYear = value ??
'',
),
SizedBox(height: 10),
_buildNumberField(
  label: 'Total Marks',
  hintText: 'e.g. 980',
  controller: _hscTotalController,
  onSave: (value) =>
    hscTotal = double.tryParse(value ??
'0') ?? 0,
),
SizedBox(height: 10),
_buildNumberField(
  label: 'Out Of',
  hintText: 'e.g. 1000',
  controller: _hscOutOfController,
  onSave: (value) =>
    hscOutOf = double.tryParse(value ??
'0') ?? 0,
),
SizedBox(height: 5),
Text(
  'Calculated Percentage:
${hscPercentage.toStringAsFixed(2)}%',
  style: TextStyle(color: Colors.indigo),
```



```
),  
Divider(height: 30, thickness: 2, color:  
Colors.orangeAccent),  
  
// SSC details  
Text('SSC Details', style: textStyle),  
SizedBox(height: 10),  
TextFormField(  
  decoration: InputDecoration(  
    labelText: 'School Name',  
    hintText: 'Enter SSC school name',  
    border: OutlineInputBorder(),  
  ),  
  onSave: (value) => sscCollege = value  
?? '',  
)  
SizedBox(height: 10),  
TextFormField(  
  decoration: InputDecoration(  
    labelText: 'Year of Passing',  
    hintText: 'YYYY',  
    border: OutlineInputBorder(),  
  ),  
  onSave: (value) => sscYear = value ??  
'',  
)  
SizedBox(height: 10),  
_buildNumberField(  
  label: 'Total Marks',  
  hintText: 'e.g. 500',  
  controller: _sscTotalController,  
  onSave: (value) =>  
    sscTotal = double.tryParse(value ??  
'0') ?? 0,  
)  
SizedBox(height: 10),  
_buildNumberField(  
  label: 'Out Of',  
  hintText: 'e.g. 600',  
  controller: _sscOutOfController,  
  onSave: (value) =>  
    sscOutOf = double.tryParse(value ??  
'0') ?? 0,  
)  
SizedBox(height: 5),  
Text(  

```



```
        'Calculated Percentage:
        ${sscPercentage.toStringAsFixed(2)}%',
        style: TextStyle(color: Colors.indigo),
    ),
    Divider(height: 30, thickness: 2, color:
Colors.orangeAccent),

    // Graduation Details Section
    Text('Graduation Details', style:
textStyle),
    SizedBox(height: 10),
    Row(
      children: [
        Text(
          'Mode: ',
          style: TextStyle(fontSize: 16,
color: Colors.indigo),
        ),
        SizedBox(width: 10),
        DropdownButton<bool>({
          value: isGraduationCGPA,
          items: [
            DropdownMenuItem(
              child: Text('CGPA'),
              value: true,
            ),
            DropdownMenuItem(
              child: Text('Percentage'),
              value: false,
            ),
          ],
          onChanged: (bool? newValue) {
            setState(() {
              isGraduationCGPA = newValue ??
true;
            });
          },
        ),
      ],
    ),
    SizedBox(height: 10),
    // Graduation marks for 8 semesters
    ListView.builder(
      shrinkWrap: true,
      physics: NeverScrollableScrollPhysics(),
      itemCount: 8,
```



```
itemBuilder: (context, index) {  
    return Column(  
        crossAxisAlignment:  
CrossAxisAlignment.start,  
        children: [  
            Text(  
                'Semester ${index + 1}'  
                (${isGraduationCGPA ? "CGPA" : "Percentage"}'),  
                style: TextStyle(color:  
Colors.indigo),  
            ),  
            SizedBox(height: 5),  
            TextFormField(  
                decoration: InputDecoration(  
                    labelText:  
                'Enter ${isGraduationCGPA  
? "CGPA" : "Percentage"}',  
                    hintText: isGraduationCGPA  
                        ? 'e.g. 8.5'  
                        : 'e.g. 85',  
                    border: OutlineInputBorder(),  
                ),  
                keyboardType:  
TextInputType.number,  
                onSave: (value) =>  
gradList[index] =  
                    double.tryParse(value ??  
'0') ?? 0,  
            ),  
            SizedBox(height: 10),  
        ],  
    );  
},  
),  
SizedBox(height: 5),  
Text(  
    'Aggregate: Will be calculated upon  
submission',  
    style: TextStyle(color: Colors.indigo),  
),  
Divider(height: 30, thickness: 2, color:  
Colors.orangeAccent),  
  
// Resume Upload  
Text('Upload Resume', style: textStyle),  
SizedBox(height: 10),
```




```
Row(  
  children: [  
    ElevatedButton.icon(  
      style: ElevatedButton.styleFrom(  
        backgroundColor:  
Colors.orangeAccent),  
      onPressed: _pickResume,  
      icon: Icon(Icons.upload_file, color:  
Colors.white),  
      label: Text('Choose File',  
        style: TextStyle(color:  
Colors.white)),  
    ),  
    SizedBox(width: 10),  
    Expanded(  
      child: Text(  
        resumeFile.isEmpty ? 'No file  
chosen' : resumeFile,  
        style: TextStyle(color:  
Colors.indigo),  
      ),  
    ),  
  ],  
,  
),  
SizedBox(height: 20),  
  
// Additional Courses  
Text('Any Additional Courses', style:  
textStyle),  
  
SizedBox(height: 10),  
TextFormField(  
  maxLines: 3,  
  decoration: InputDecoration(  
    labelText: 'Additional Courses',  
    hintText: 'List any additional courses  
you have taken',  
    border: OutlineInputBorder(),  
  ),  
  onSave: (value) => additionalCourses =  
value ?? '',  
,  
),  
SizedBox(height: 20),  
  
ElevatedButton(  
  style: ElevatedButton.styleFrom(  
    backgroundColor: Colors.indigo,
```



```
padding:
EdgeInsets.symmetric(vertical: 16),
    textStyle: TextStyle(fontSize: 18),
  ),
  onPressed: _submitForm,
  child: Text(
    'Submit',
    style:
      TextStyle(color:
Color.fromARGB(255, 244, 242, 241)),
  ),
),

// Button to view the uploaded resume
(shows only the name)
if (resumeFile.isNotEmpty)
  TextButton(
    onPressed: () {
      Navigator.pushNamed(context,
'/resumeView',
        arguments: resumeFile);
    },
    child: Text('View Uploaded Resume',
      style:
        TextStyle(color:
Colors.indigo, fontSize: 16)),
  ),
],
),
),
),
),
),
);
}
}

// Company Details Page with updated companies.
class CompanyDetailsPage extends StatelessWidget {
  // Example companies with random info.
  final List<Map<String, String>> companies = [
    {
      'name': 'Google',
      'location': 'Mountain View, CA',
      'salaryRange': '20-40 LPA',
      'profile': 'Leading innovator in search and cloud
services.'
```



```
    },  
    {  
      'name': 'Amazon',  
      'location': 'Seattle, WA',  
      'salaryRange': '18-35 LPA',  
      'profile': 'E-commerce giant with extensive cloud  
and AI services.'  
    },  
    {  
      'name': 'Microsoft',  
      'location': 'Redmond, WA',  
      'salaryRange': '22-38 LPA',  
      'profile': 'Pioneer in software, hardware, and cloud  
solutions.'  
    },  
  ],  
];  
  
@override  
Widget build(BuildContext context) {  
  return ListView.builder(  
    padding: EdgeInsets.all(16),  
    itemCount: companies.length,  
    itemBuilder: (context, index) {  
      final company = companies[index];  
      return Card(  
        elevation: 4,  
        margin: EdgeInsets.only(bottom: 16),  
        shape:  
          RoundedRectangleBorder(borderRadius:  
BorderRadius.circular(12)),  
        child: ListTile(  
          title: Text(company['name']!,  
            style: TextStyle(  
              color: Colors.indigo, fontWeight:  
FontWeight.bold)),  
          subtitle: Column(  
            crossAxisAlignment:  
CrossAxisAlignment.start,  
            children: [  
              Text('Location: ${company['location']}',  
                style: TextStyle(color:  
Colors.black87)),  
              Text('Salary Range:  
${company['salaryRange']}',  
                style: TextStyle(color:  
Colors.black87)),  
            ],  
          ),  
        ),  
      );  
    },  
  );  
}
```



```
Text('Profile: ${company['profile']}',  
      style: TextStyle(color:  
Colors.black87)),  
    ],  
  ),  
),  
);  
},  
);  
}  
}
```

// My Details Page (for student to view own details)
showing saved registration data.

```
class MyDetailsPage extends StatelessWidget {  
  @override  
  Widget build(BuildContext context) {  
    // If registrationData is null, then no data was  
    saved.  
    return Padding(  
      padding: EdgeInsets.all(16),  
      child: Card(  
        elevation: 6,  
        color: Colors.white,  
        shape: RoundedRectangleBorder(borderRadius:  
BorderRadius.circular(16)),  
        child: Padding(  
          padding: EdgeInsets.all(20),  
          child: registrationData != null  
            ? SingleChildScrollView(  
              child: Column(  
                crossAxisAlignment:  
CrossAxisAlignment.start,  
                children: [  
                  Text('My Details',  
                    style: TextStyle(  
                      fontSize: 24,  
                      fontWeight: FontWeight.bold,  
                      color: Colors.indigo)),  
                  Divider(color: Colors.orangeAccent,  
thickness: 2),  
                  Text('Name:  
${registrationData!.name}',  
                    style: TextStyle(fontSize: 18)),  
                  Text('Email:  
${registrationData!.email}',
```



```
style: TextStyle(fontSize: 18)),
Text('Contact:
${registrationData!.contact}',
style: TextStyle(fontSize: 18)),
Text('Roll No:
${registrationData!.rollNo}',
style: TextStyle(fontSize: 18)),
SizedBox(height: 10),
Text(
'HSC:
${registrationData!.hscCollege}
(${registrationData!.hscYear}) -
${registrationData!.hscPercentage.toStringAsFixed(2)}%',
style: TextStyle(fontSize: 16)),
Text(
'SSC:
${registrationData!.sscCollege}
(${registrationData!.sscYear}) -
${registrationData!.sscPercentage.toStringAsFixed(2)}%',
style: TextStyle(fontSize: 16)),
SizedBox(height: 10),
Text(
'Graduation
(${registrationData!.isGraduationCGPA ? "CGPA" :
"Percentage"}): Aggregate -
${registrationData!.gradAggregate.toStringAsFixed(2)}',
style: TextStyle(fontSize: 16)),
SizedBox(height: 10),
Text('Resume:
${registrationData!.resumeFile}',
style: TextStyle(fontSize: 16)),
SizedBox(height: 10),
Text(
'Additional Courses:
${registrationData!.additionalCourses}',
style: TextStyle(fontSize: 16)),
],
),
)
: Center(
child: Text(
'No registration details
found.\nPlease fill in the registration form.',
textAlign: TextAlign.center,
style: TextStyle(fontSize: 18, color:
Colors.indigo),
```

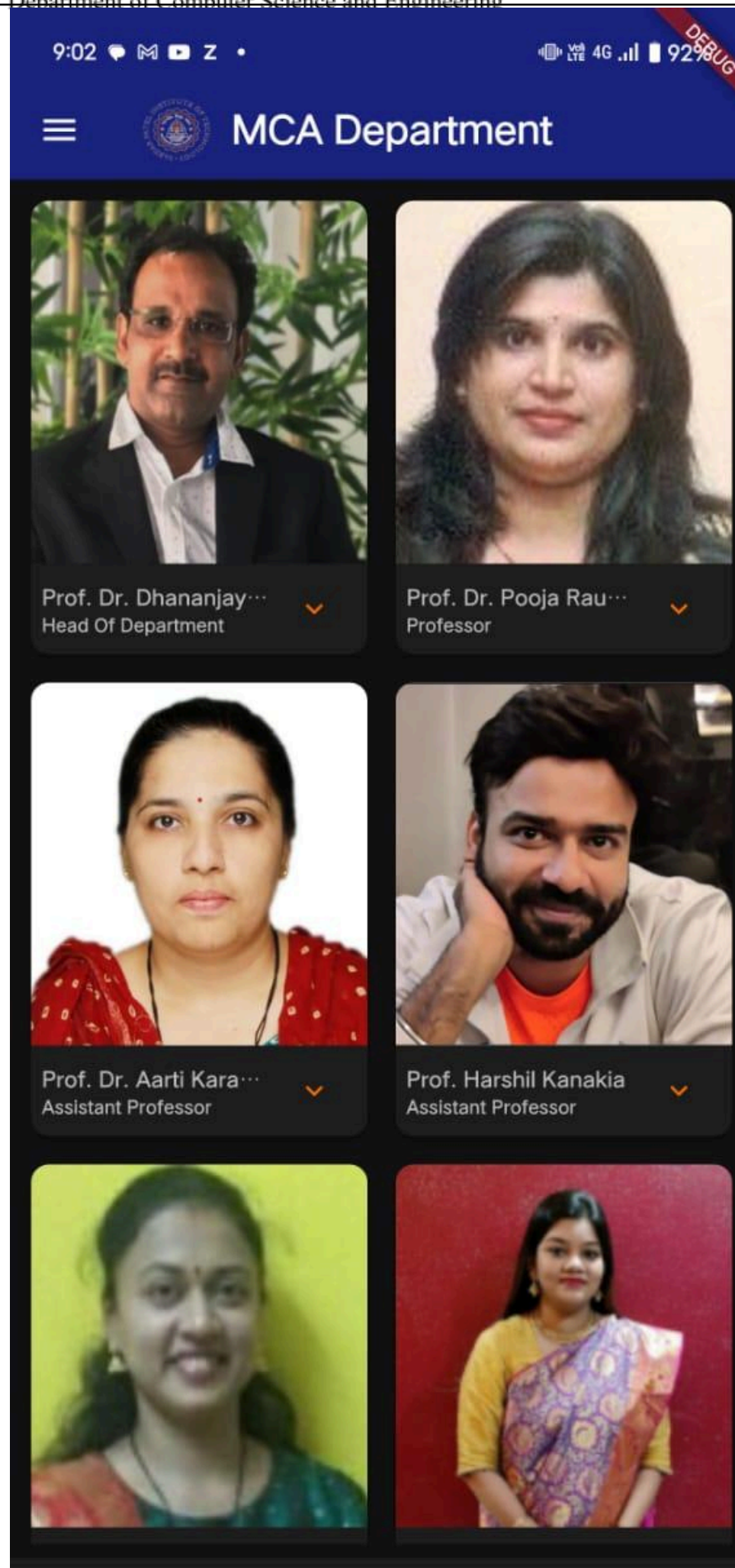


```
    ),  
    ),  
    ),  
    ),  
);  
}  
}  
  
// Resume View Page (Optional) - Displays only the file  
name.  
class ResumeViewPage extends StatelessWidget {  
  @override  
  Widget build(BuildContext context) {  
    // The resume file name is passed as an argument.  
    final resumeFile =  
      ModalRoute.of(context)!.settings.arguments as  
String? ?? 'No Resume';  
  
    return Scaffold(  
      appBar: AppBar(  
        title: Text('View Resume'),  
        backgroundColor: Colors.indigo,  
      ),  
      body: Center(  
        child: Card(  
          elevation: 4,  
          color: Colors.white,  
          margin: EdgeInsets.all(16),  
          shape:  
            RoundedRectangleBorder(borderRadius:  
BorderRadius.circular(12)),  
          child: Padding(  
            padding: const EdgeInsets.all(20),  
            child: Text(  
              'Uploaded Resume: $resumeFile',  
              style: TextStyle(fontSize: 20, color:  
Colors.indigo),  
              textAlign: TextAlign.center,  
            ),  
          ),  
        ),  
      ),  
    );  
  }  
}
```



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

Screenshots





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~~

--	--



9:02 •

92%



MCA Department

Department Vision

Our vision is to become a premier center of excellence in IT education and research. We strive to create an environment where students and faculty collaborate to innovate, adapt, and excel in the rapidly evolving world of technology.

Department Mission

The mission of the MCA Department is to provide high-quality education that balances theoretical knowledge with practical skills. We are committed to:

1. Delivering a curriculum that is current with industry trends and technological advancements.
2. Fostering an innovative and research-oriented environment.
3. Promoting interdisciplinary learning and collaboration.
4. Equipping students with the skills needed for successful careers in software engineering, data science, and IT management.
5. Encouraging lifelong learning and professional growth among our graduates.



9:03

MCA Department

Placement Update: Top Recruiters Visit on April 10, 2025

Our placement cell has arranged campus interviews with leading IT companies. Check details for schedules and preparation tips.

Webinar on Emerging Technologies - Register Now!

Join experts as they discuss trends in AI, IoT, and blockchain. Don't miss this opportunity to enhance your skills.

Annual Fest 2025: Schedule and Events Announced

Get ready for an exciting array of competitions, workshops, and cultural events at the MCA Annual Fest 2025.

Admissions Open for MCA 2025 Batch

Apply now to be a part of one of the most sought-after MCA programs. Check eligibility criteria and application process.



9:03

MCA Department

MCA Website

**Master of
Computer
Applications**

**Welcome to MCA
Department of Sardar
Patel Institute of
Technology**

Right from it's first year in 2009, S.P.I.T. has been one of the sought college by students for MCA and is renown for its ongoing courses. Even though starting in year 2009 it has managed to excel quickly and with the highest cutoff this year it has shown a growth among the students all over.



9:03

92%

MCA Department

Rate our Department

Your Feedback

Best College

Submit

Feedback submitted! Rating: 4.0, Feedback: Best College



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

Question and Answers	<p>Answer the following Questions:</p> <ol style="list-style-type: none">1. How to create SideDrawer and Tabs Navigation both? SideDrawer navigation is created by adding a Drawer widget to the Scaffold, which includes a DrawerHeader and ListTile items for each navigation option. Tabs navigation can be implemented by using a DefaultTabController along with a TabBar and a TabBarView, allowing users to switch between different views within the same screen.2. How did you use 60-30-10 rule in your application? The 60-30-10 rule is applied through the color scheme of the app. The dominant color takes up about 60% of the design, the secondary color covers around 30%, and the accent color is used sparingly for about 10% to highlight key UI elements, ensuring a balanced and visually appealing interface.3. Which new elements did you use for creating UI components? New UI components used in the application include Card widgets for grouping content, ListView.builder for dynamic lists, Drawer for side navigation, and buttons like ElevatedButton and TextButton for user interaction. The use of FilePicker for resume uploads also adds a modern touch to the user interface.4. In pubspec.yaml file, what dependencies need to be there? In the pubspec.yaml file, dependencies such as flutter, cupertino_icons, and file_picker are necessary. Additional dependencies might include packages for webview functionality (like webview_flutter) if web content needs to be displayed within the app.5. How to create Expandable view? An expandable view can be created using the ExpansionTile widget, which provides a built-in way to show and hide content based on user interaction. This makes it easier to manage and display large amounts of content in a compact space.6. How to create Webview? Creating a webview in Flutter is possible with packages like webview_flutter. By adding the package to your pubspec.yaml file and importing it in your code, you can embed a WebView widget to display web content within your app.
Conclusion	<p>In conclusion, the application demonstrates a blend of side drawer and tab navigation techniques, uses a well-structured 60-30-10 color strategy, integrates modern UI components and essential dependencies, and also hints at expandable views and webview capabilities for a comprehensive and user-friendly experience.</p>