|  |  |
| --- | --- |
| **Experiment** | 4 |
| **Aim** | Develop an app for TPO Office at SPIT |
| **Objective** | * To Create TPO App using UI Component * To Develop SideDrawer Navigation * To Use 60-30-10 rule |
| **Name** | Aditya Reddy |
| **UCID** | 2024510049 |
| **Class** | FY MCA |
| **Batch** | C |
| **Date of Submission** | 22-02-2025 |

|  |  |
| --- | --- |
| **Technology used** | Flutter, Dart |
| **Task** | Develop an app for TPO with the following requirement (Use side navigation drawer) :   1. Registration Form (Name, E-mail id, Contact, Roll no, HSC (Which College, Year of Passing, total, out of, should calculate percentage),SSC marks (Which College, Year of Passing, total, out of, should calculate percentage), Marks in sem 1-5 (CGPA and Percentage), upload Resume, Any Additional Courses). 2. User should be able to view the uploaded Resume. (optional) 3. Company details should be visible to the user. (Name of the company, Location, Range of the salary be given for internship/Placement, Short Profile). 4. Student should be able to view their own details. |
| **Code with proper label** | **register\_screen.dart**  import 'package:flutter/material.dart'; import 'package:file\_picker/file\_picker.dart';  class RegisterScreen extends StatefulWidget {  final Function(Map<String, dynamic>) updateStudentData;  RegisterScreen({required this.updateStudentData});   @override  \_RegisterScreenState createState() => \_RegisterScreenState(); }  class \_RegisterScreenState extends State<RegisterScreen> {  final \_formKey = GlobalKey<FormState>();  final Map<String, dynamic> \_formData = {};  String? \_resumeFileName;  String? \_resumeFilePath;   double calculatePercentage(int obtained, int total) {  return (obtained / total) \* 100;  }   Future<void> \_pickResume() async {  FilePickerResult? result = await FilePicker.*platform*.pickFiles(type: FileType.custom, allowedExtensions: ['pdf']);  if (result != null) {  setState(() {  \_resumeFileName = result.files.single.name;  \_resumeFilePath = result.files.single.path;  \_formData['resumePath'] = \_resumeFilePath;  });  }  }   void \_showSnackbar(BuildContext context) {  ScaffoldMessenger.*of*(context).showSnackBar(  SnackBar(  content: Text('Form Submitted Successfully!'),  backgroundColor: Colors.*green*,  ),  );  }   @override  Widget build(BuildContext context) {  return Scaffold(  appBar: AppBar(title: Text('Register')),  body: Padding(  padding: const EdgeInsets.all(16.0),  child: SingleChildScrollView(  child: Form(  key: \_formKey,  child: Column(  children: <Widget>[  TextFormField(decoration: InputDecoration(labelText: 'Name'), validator: (val) => val!.isEmpty ? 'Required' : null, onSaved: (val) => \_formData['name'] = val),  TextFormField(decoration: InputDecoration(labelText: 'Email'), validator: (val) => val!.isEmpty ? 'Required' : null, onSaved: (val) => \_formData['email'] = val),  TextFormField(decoration: InputDecoration(labelText: 'Contact'), keyboardType: TextInputType.*number*, validator: (val) => val!.isEmpty ? 'Required' : null, onSaved: (val) => \_formData['contact'] = val),  TextFormField(decoration: InputDecoration(labelText: 'Roll No'), keyboardType: TextInputType.*number*, validator: (val) => val!.isEmpty ? 'Required' : null, onSaved: (val) => \_formData['rollNo'] = val),  TextFormField(decoration: InputDecoration(labelText: 'Graduation College'), validator: (val) => val!.isEmpty ? 'Required' : null, onSaved: (val) => \_formData['gradCollege'] = val),  TextFormField(decoration: InputDecoration(labelText: 'Graduation Year of Passing'), keyboardType: TextInputType.*number*, validator: (val) => val!.isEmpty ? 'Required' : null, onSaved: (val) => \_formData['gradYear'] = val),  TextFormField(decoration: InputDecoration(labelText: 'HSC College'), validator: (val) => val!.isEmpty ? 'Required' : null, onSaved: (val) => \_formData['hscCollege'] = val),  TextFormField(decoration: InputDecoration(labelText: 'HSC Year of Passing'), keyboardType: TextInputType.*number*, validator: (val) => val!.isEmpty ? 'Required' : null, onSaved: (val) => \_formData['hscYear'] = val),  TextFormField(decoration: InputDecoration(labelText: 'HSC Total'), keyboardType: TextInputType.*number*, validator: (val) => val!.isEmpty ? 'Required' : null, onSaved: (val) => \_formData['hscTotal'] = int.*parse*(val!)),  TextFormField(decoration: InputDecoration(labelText: 'HSC Out Of'), keyboardType: TextInputType.*number*, validator: (val) => val!.isEmpty ? 'Required' : null, onSaved: (val) => \_formData['hscOutOf'] = int.*parse*(val!)),  TextFormField(decoration: InputDecoration(labelText: 'SSC College'), validator: (val) => val!.isEmpty ? 'Required' : null, onSaved: (val) => \_formData['sscCollege'] = val),  TextFormField(decoration: InputDecoration(labelText: 'SSC Year of Passing'), keyboardType: TextInputType.*number*, validator: (val) => val!.isEmpty ? 'Required' : null, onSaved: (val) => \_formData['sscYear'] = val),  TextFormField(decoration: InputDecoration(labelText: 'SSC Total'), keyboardType: TextInputType.*number*, validator: (val) => val!.isEmpty ? 'Required' : null, onSaved: (val) => \_formData['sscTotal'] = int.*parse*(val!)),  TextFormField(decoration: InputDecoration(labelText: 'SSC Out Of'), keyboardType: TextInputType.*number*, validator: (val) => val!.isEmpty ? 'Required' : null, onSaved: (val) => \_formData['sscOutOf'] = int.*parse*(val!)),  TextFormField(  decoration: InputDecoration(labelText: 'Marks in Sem 1 (CGPA)'),  keyboardType: TextInputType.*number*,  validator: (val) => val!.isEmpty ? 'Required' : null,  onSaved: (val) => \_formData['sem1'] = double.*tryParse*(val!),  ),  TextFormField(  decoration: InputDecoration(labelText: 'Marks in Sem 2 (CGPA)'),  keyboardType: TextInputType.*number*,  validator: (val) => val!.isEmpty ? 'Required' : null,  onSaved: (val) => \_formData['sem2'] = double.*tryParse*(val!),  ),  TextFormField(  decoration: InputDecoration(labelText: 'Marks in Sem 3 (CGPA)'),  keyboardType: TextInputType.*number*,  validator: (val) => val!.isEmpty ? 'Required' : null,  onSaved: (val) => \_formData['sem3'] = double.*tryParse*(val!),  ),  TextFormField(  decoration: InputDecoration(labelText: 'Marks in Sem 4 (CGPA)'),  keyboardType: TextInputType.*number*,  validator: (val) => val!.isEmpty ? 'Required' : null,  onSaved: (val) => \_formData['sem4'] = double.*tryParse*(val!),  ),  TextFormField(  decoration: InputDecoration(labelText: 'Marks in Sem 5 (CGPA)'),  keyboardType: TextInputType.*number*,  validator: (val) => val!.isEmpty ? 'Required' : null,  onSaved: (val) => \_formData['sem5'] = double.*tryParse*(val!),  ),  TextFormField(decoration: InputDecoration(labelText: 'Additional Courses'), onSaved: (val) => \_formData['courses'] = val),  ElevatedButton(  onPressed: \_pickResume,  child: Text('Upload Resume (PDF)'),  ),  if (\_resumeFileName != null) Text('Selected file: $\_resumeFileName', style: TextStyle(color: Colors.*green*)),  ElevatedButton(  onPressed: () {  if (\_formKey.currentState!.validate()) {  \_formKey.currentState!.save();  \_formData['hscPercentage'] = calculatePercentage(\_formData['hscTotal'], \_formData['hscOutOf']);  \_formData['sscPercentage'] = calculatePercentage(\_formData['sscTotal'], \_formData['sscOutOf']);  widget.updateStudentData(\_formData);  \_showSnackbar(context);  }  },  child: Text('Submit'),  ),  ],  ),  ),  ),  ),  );  } }  **view\_resume\_screen.dart**  import 'package:flutter/material.dart'; import 'package:open\_file/open\_file.dart';  class ViewResumeScreen extends StatelessWidget {  final String? resumeFilePath;  ViewResumeScreen({this.resumeFilePath});   void \_openResume() {  if (resumeFilePath != null) {  OpenFile.*open*(resumeFilePath);  }  }   @override  Widget build(BuildContext context) {  return Scaffold(  appBar: AppBar(title: Text('View Resume')),  body: Center(  child: resumeFilePath != null  ? Column(  mainAxisAlignment: MainAxisAlignment.center,  children: [  Text('Resume Uploaded', style: TextStyle(fontSize: 18, fontWeight: FontWeight.*bold*)),  SizedBox(height: 10),  ElevatedButton(  onPressed: \_openResume,  child: Text('Open Resume'),  ),  ],  )  : Text('No Resume Uploaded', style: TextStyle(color: Colors.*red*)),  ),  );  } }  **company\_details.dart**  import 'package:flutter/material.dart';  class CompanyDetailsScreen extends StatelessWidget {  @override  Widget build(BuildContext context) {  return Scaffold(  appBar: AppBar(  title: Text('Company Details'),  ),  body: ListView(  padding: const EdgeInsets.all(8),  children: <Widget>[  ListTile(  title: Text('Company Name: PhonePe'),  subtitle: Text(''' Location: Bangalore Salary Range: ₹15,00,000 - ₹40,00,000 (Placement) Profile: Technology and Software Development'''),  ),  ListTile(  title: Text('Company Name: Microsoft'),  subtitle: Text(''' Location: Hyderabad Salary Range: ₹14,00,000 - ₹35,00,000 (Placement) Profile: Software Development, Cloud Computing, AI, and Enterprise Solutions'''),  ),  ListTile(  title: Text('Company Name: Amazon'),  subtitle: Text(''' Location: Chennai Salary Range: ₹12,00,000 - ₹30,00,000 (Placement) Profile: E-Commerce, Cloud Services, AI, and Logistics'''),  ),  ListTile(  title: Text('Company Name: Infosys'),  subtitle: Text(''' Location: Pune Salary Range: ₹20,000 - ₹50,000 (Internship) Profile: IT Consulting, Software Development, and AI Solutions'''),  ),  ListTile(  title: Text('Company Name: Deloitte'),  subtitle: Text(''' Location: Gurugram Salary Range: ₹6,00,000 - ₹15,00,000 (Placement) Profile: Consulting, Finance, IT Solutions, and Business Services'''),  ),  ],  ),  );  } }  **student\_details.dart**  import 'package:flutter/material.dart';  class StudentDetailsScreen extends StatelessWidget {  final Map<String, dynamic> studentData;  StudentDetailsScreen({required this.studentData});   @override  Widget build(BuildContext context) {  return Scaffold(  appBar: AppBar(title: Text('Student Details')),  body: Padding(  padding: const EdgeInsets.all(16.0),  child: SingleChildScrollView(  child: Column(  crossAxisAlignment: CrossAxisAlignment.start,  children: <Widget>[  Text('Name: ${studentData['name'] ?? "N/A"}'),  Text('Email: ${studentData['email'] ?? "N/A"}'),  Text('Contact: ${studentData['contact'] ?? "N/A"}'),  Text('Roll No: ${studentData['rollNo'] ?? "N/A"}'),  Text('HSC Percentage: ${studentData['hscPercentage']?.toStringAsFixed(2) ?? "N/A"}%'),  Text('SSC Percentage: ${studentData['sscPercentage']?.toStringAsFixed(2) ?? "N/A"}%'),  Text('Marks in Sem 1: ${studentData['sem1'] ?? "N/A"} CGPA'),  Text('Marks in Sem 2: ${studentData['sem2'] ?? "N/A"} CGPA'),  Text('Marks in Sem 3: ${studentData['sem3'] ?? "N/A"} CGPA'),  Text('Marks in Sem 4: ${studentData['sem4'] ?? "N/A"} CGPA'),  Text('Marks in Sem 5: ${studentData['sem5'] ?? "N/A"} CGPA'),  Text('Additional Courses: ${studentData['courses'] ?? "N/A"}'),  ],  ),  ),  ),  );  } } |
| **Screenshots** | Registration:    Validation on registration:    View resume:      Company details:    Student details:    SideNav(Drawer): |
| **Question and Answers** | Answer the following Questions:   1. How to create SideDrawer Navigation?   SideDrawer Navigation can be created using a drawer component that slides in from the side when a button (like a hamburger menu) is clicked. Libraries like Material-UI or Tailwind with Headless UI make this easy by providing built-in drawer components.   1. How did you use 60-30-10 rule in your application?   60-30-10 Rule is a color scheme principle used for a balanced UI. In my application, 60% is the primary color (background or main sections), 30% is the secondary color (buttons or sidebars), and 10% is the accent color (highlights or call-to-action buttons), ensuring a visually appealing design.   1. Which new elements did you use for creating UI components?   New UI Elements used include cards for grouped information, modals for pop-ups, tabs for content organization, tooltips for additional details, and progress bars for tracking progress. These elements enhance usability and improve the user experience. |
| **Conclusion** | This project is a **Training and Placement Office (TPO) App** designed to help students manage their registration, resumes, and academic details in one place. The app allows students to fill out their information, upload their resumes, and view company details.  Learned new things like create side navigation with a Drawer, handle forms and validation, and manage file uploads using file\_picker and open\_file. |