|  |  |
| --- | --- |
| **Experiment** | LAB ESE Submission |
| **Aim** | Create Scholarship management App. This system for the needy people especially family who are living under below poverty line can apply and will get 15000l per annum for their ward. It makes the whole process of the system easy by providing the easier application filling, payment receiving system. |
| **Name** | Aditya Reddy |
| **UCID** | 2024510049 |
| **Class** | MCA |
| **Batch** | C |
| **Date of Examination** | 12/05/2025 |

|  |  |
| --- | --- |
| **Technology used** | Flutter, Dart, Android Studio |
| **Task** | Create a Scholorship Management App for easier form filling and payment process |
| **Code with proper label** | Main.dart import 'package:flutter/material.dart'; import 'package:provider/provider.dart'; import 'package:google\_fonts/google\_fonts.dart'; import 'screens/home\_screen.dart'; import 'screens/application\_form.dart'; import 'screens/status\_screen.dart'; import 'providers/scholarship\_provider.dart';  void main() {  runApp(const MyApp()); }  class MyApp extends StatelessWidget {  const MyApp({super.key});   @override  Widget build(BuildContext context) {  return ChangeNotifierProvider(  create: (ctx) => ScholarshipProvider(),  child: MaterialApp(  debugShowCheckedModeBanner: false,  title: 'MahaDBT Scholarship',  theme: ThemeData(  // 60% - Primary background color  scaffoldBackgroundColor: const Color(0xFF1A1A1A),    // Modern typography  textTheme: GoogleFonts.*interTextTheme*(  Theme.*of*(context).textTheme,  ).copyWith(  // Headings  headlineLarge: GoogleFonts.*inter*(  fontSize: 32,  fontWeight: FontWeight.*bold*,  color: Colors.*white*,  ),  headlineMedium: GoogleFonts.*inter*(  fontSize: 24,  fontWeight: FontWeight.*bold*,  color: Colors.*white*,  ),  // Body text  bodyLarge: GoogleFonts.*inter*(  fontSize: 16,  color: Colors.*white70*,  ),  bodyMedium: GoogleFonts.*inter*(  fontSize: 14,  color: Colors.*white70*,  ),  ),   // 30% - Secondary color  primaryColor: const Color(0xFF2D2D2D),    // Input decoration theme  inputDecorationTheme: InputDecorationTheme(  filled: true,  fillColor: const Color(0xFF2D2D2D),  border: OutlineInputBorder(  borderRadius: BorderRadius.circular(8),  borderSide: const BorderSide(color: Color(0xFF404040)),  ),  enabledBorder: OutlineInputBorder(  borderRadius: BorderRadius.circular(8),  borderSide: const BorderSide(color: Color(0xFF404040)),  ),  focusedBorder: OutlineInputBorder(  borderRadius: BorderRadius.circular(8),  borderSide: const BorderSide(color: Color(0xFF6C63FF), width: 2),  ),  labelStyle: const TextStyle(color: Colors.*white70*),  contentPadding: const EdgeInsets.symmetric(horizontal: 16, vertical: 16),  ),   // Card theme  cardTheme: CardTheme(  elevation: 4,  shape: RoundedRectangleBorder(  borderRadius: BorderRadius.circular(12),  ),  color: const Color(0xFF2D2D2D),  ),   // 10% - Accent color for buttons  elevatedButtonTheme: ElevatedButtonThemeData(  style: ElevatedButton.*styleFrom*(  backgroundColor: const Color(0xFF6C63FF),  foregroundColor: Colors.*white*,  elevation: 4,  padding: const EdgeInsets.symmetric(horizontal: 24, vertical: 16),  shape: RoundedRectangleBorder(  borderRadius: BorderRadius.circular(8),  ),  ),  ),   // AppBar theme  appBarTheme: const AppBarTheme(  backgroundColor: Color(0xFF2D2D2D),  foregroundColor: Colors.*white*,  elevation: 0,  centerTitle: true,  ),   // Bottom Navigation Bar theme  bottomNavigationBarTheme: const BottomNavigationBarThemeData(  backgroundColor: Color(0xFF2D2D2D),  selectedItemColor: Color(0xFF6C63FF),  unselectedItemColor: Colors.*white70*,  type: BottomNavigationBarType.fixed,  elevation: 8,  ),   // Dropdown theme  dropdownMenuTheme: const DropdownMenuThemeData(  textStyle: TextStyle(color: Colors.*white*),  ),  ),  home: const MainScreen(),  ),  );  } }  class MainScreen extends StatefulWidget {  const MainScreen({super.key});   @override  State<MainScreen> createState() => \_MainScreenState(); }  class \_MainScreenState extends State<MainScreen> {  int \_selectedIndex = 0;    final List<Widget> \_screens = [  const HomeScreen(),  const ApplicationForm(),  const StatusScreen(),  ];   void \_onItemTapped(int index) {  setState(() {  \_selectedIndex = index;  });  }   @override  Widget build(BuildContext context) {  return Scaffold(  body: \_screens[\_selectedIndex],  bottomNavigationBar: BottomNavigationBar(  items: const <BottomNavigationBarItem>[  BottomNavigationBarItem(  icon: Icon(Icons.*home\_rounded*),  label: 'Home',  ),  BottomNavigationBarItem(  icon: Icon(Icons.*edit\_document*),  label: 'Apply',  ),  BottomNavigationBarItem(  icon: Icon(Icons.*check\_circle\_outline*),  label: 'Status',  ),  ],  currentIndex: \_selectedIndex,  onTap: \_onItemTapped,  ),  );  } }  home\_screen.dart  import 'package:flutter/material.dart';  class HomeScreen extends StatelessWidget {  const HomeScreen({super.key});   @override  Widget build(BuildContext context) {  return Scaffold(  appBar: AppBar(  title: const Text(  'MahaDBT Scholarship',  style: TextStyle(fontWeight: FontWeight.*bold*),  ),  ),  body: SingleChildScrollView(  padding: const EdgeInsets.all(24.0),  child: Column(  crossAxisAlignment: CrossAxisAlignment.start,  children: [  Text(  'Welcome to MahaDBT\nScholarship Portal',  style: Theme.*of*(context).textTheme.headlineLarge,  ),  const SizedBox(height: 32),  Card(  child: Padding(  padding: const EdgeInsets.all(24.0),  child: Column(  crossAxisAlignment: CrossAxisAlignment.start,  children: [  Text(  'About the Scholarship',  style: Theme.*of*(context).textTheme.headlineMedium,  ),  const SizedBox(height: 16),  Text(  'This scholarship program provides financial assistance of ₹1,50,000 per annum to deserving students from economically weaker sections.',  style: Theme.*of*(context).textTheme.bodyLarge,  ),  ],  ),  ),  ),  const SizedBox(height: 24),  Card(  child: Padding(  padding: const EdgeInsets.all(24.0),  child: Column(  crossAxisAlignment: CrossAxisAlignment.start,  children: [  Text(  'Eligibility Criteria',  style: Theme.*of*(context).textTheme.headlineMedium,  ),  const SizedBox(height: 16),  \_buildEligibilityCriterion(  context,  'Family income should be below ₹8,00,000 per annum',  Icons.*currency\_rupee\_rounded*,  ),  \_buildEligibilityCriterion(  context,  'Must belong to SC/ST/OBC category',  Icons.*people\_outline\_rounded*,  ),  \_buildEligibilityCriterion(  context,  'Must be a resident of Maharashtra',  Icons.*location\_on\_outlined*,  ),  \_buildEligibilityCriterion(  context,  'Must be enrolled in a recognized educational institution',  Icons.*school\_outlined*,  ),  ],  ),  ),  ),  const SizedBox(height: 32),  SizedBox(  width: double.*infinity*,  child: ElevatedButton(  onPressed: () {  DefaultTabController.*of*(context).animateTo(1);  },  child: const Text(  'Apply Now',  style: TextStyle(  fontSize: 18,  fontWeight: FontWeight.*bold*,  ),  ),  ),  ),  ],  ),  ),  );  }   Widget \_buildEligibilityCriterion(BuildContext context, String text, IconData icon) {  return Padding(  padding: const EdgeInsets.only(bottom: 16.0),  child: Row(  children: [  Icon(  icon,  color: const Color(0xFF6C63FF),  size: 24,  ),  const SizedBox(width: 16),  Expanded(  child: Text(  text,  style: Theme.*of*(context).textTheme.bodyLarge,  ),  ),  ],  ),  );  } }  application\_form.dart import 'package:flutter/material.dart'; import 'package:provider/provider.dart'; import 'package:intl/intl.dart'; import '../providers/scholarship\_provider.dart';  class ApplicationForm extends StatefulWidget {  const ApplicationForm({super.key});   @override  State<ApplicationForm> createState() => \_ApplicationFormState(); }  class \_ApplicationFormState extends State<ApplicationForm> {  final \_formKey = GlobalKey<FormState>();  final Map<String, String> \_formData = {  'fullName': '',  'dateOfBirth': '',  'gender': '',  'caste': '',  'familyIncome': '',  'address': '',  'district': '',  'aadharNumber': '',  'bankAccount': '',  'ifscCode': '',  'institutionName': '',  'courseName': '',  };   final List<String> \_districts = [  'Mumbai',  'Pune',  'Nagpur',  'Thane',  'Nashik',  'Aurangabad',  'Solapur',  'Amravati',  'Kolhapur',  'Other'  ];   final List<String> \_castes = [  'Open',  'OBC',  'SC',  'ST',  'Other'  ];   final List<String> \_courses = [  'B-Tech CS',  'B-Tech IT',  'B-Tech AI/ML',  'MCA',  'M-Tech'  ];   bool \_isValidAge(DateTime date) {  final age = DateTime.now().difference(date).inDays ~/ 365;  return age >= 14;  }   bool \_isValidIFSC(String ifsc) {  final ifscRegex = RegExp(r'^[A-Z]{3}[0-9]{4}$');  return ifscRegex.hasMatch(ifsc);  }   Future<void> \_selectDate(BuildContext context) async {  final DateTime? picked = await showDatePicker(  context: context,  initialDate: DateTime.now().subtract(const Duration(days: 365 \* 14)),  firstDate: DateTime(1990),  lastDate: DateTime.now(),  builder: (context, child) {  return Theme(  data: Theme.*of*(context).copyWith(  colorScheme: const ColorScheme.dark(  primary: Color(0xFF6C63FF),  onPrimary: Colors.*white*,  surface: Color(0xFF2D2D2D),  onSurface: Colors.*white*,  ),  ),  child: child!,  );  },  );  if (picked != null) {  if (\_isValidAge(picked)) {  setState(() {  \_formData['dateOfBirth'] = DateFormat('dd/MM/yyyy').format(picked);  });  } else {  ScaffoldMessenger.*of*(context).showSnackBar(  const SnackBar(  content: Text('Age must be at least 14 years'),  backgroundColor: Color(0xFFE53935),  ),  );  }  }  }   void \_submitForm() {  if (\_formKey.currentState!.validate()) {  \_formKey.currentState!.save();  Provider.*of*<ScholarshipProvider>(context, listen: false)  .submitApplication(\_formData);  Navigator.*pushReplacementNamed*(context, '/status');  }  }   @override  Widget build(BuildContext context) {  return Scaffold(  appBar: AppBar(  title: const Text('Scholarship Application'),  ),  body: SingleChildScrollView(  padding: const EdgeInsets.all(16.0),  child: Form(  key: \_formKey,  child: Column(  crossAxisAlignment: CrossAxisAlignment.stretch,  children: [  TextFormField(  decoration: const InputDecoration(  labelText: 'Full Name',  border: OutlineInputBorder(),  ),  style: const TextStyle(color: Colors.*white*),  validator: (value) {  if (value == null || value.isEmpty) {  return 'Please enter your full name';  }  if (value.length > 26) {  return 'Name should not exceed 26 characters';  }  if (!RegExp(r'^[a-zA-Z\s]+$').hasMatch(value)) {  return 'Name should contain only letters and spaces';  }  return null;  },  onSaved: (value) => \_formData['fullName'] = value!,  ),  const SizedBox(height: 16),  InkWell(  onTap: () => \_selectDate(context),  child: InputDecorator(  decoration: const InputDecoration(  labelText: 'Date of Birth',  border: OutlineInputBorder(),  ),  child: Text(  \_formData['dateOfBirth'] ?? 'Select Date',  style: TextStyle(  color: \_formData['dateOfBirth'] == null ? Colors.*white70* : Colors.*white*,  ),  ),  ),  ),  const SizedBox(height: 16),  DropdownButtonFormField<String>(  decoration: const InputDecoration(  labelText: 'Gender',  border: OutlineInputBorder(),  ),  dropdownColor: const Color(0xFF2D2D2D),  style: const TextStyle(color: Colors.*white*),  items: ['Male', 'Female', 'Other']  .map((gender) => DropdownMenuItem(  value: gender,  child: Text(gender),  ))  .toList(),  validator: (value) {  if (value == null || value.isEmpty) {  return 'Please select your gender';  }  return null;  },  onChanged: (value) => \_formData['gender'] = value!,  ),  const SizedBox(height: 16),  DropdownButtonFormField<String>(  decoration: const InputDecoration(  labelText: 'Caste',  border: OutlineInputBorder(),  ),  dropdownColor: const Color(0xFF2D2D2D),  style: const TextStyle(color: Colors.*white*),  items: \_castes  .map((caste) => DropdownMenuItem(  value: caste,  child: Text(caste),  ))  .toList(),  validator: (value) {  if (value == null || value.isEmpty) {  return 'Please select your caste';  }  return null;  },  onChanged: (value) => \_formData['caste'] = value!,  ),  const SizedBox(height: 16),  TextFormField(  decoration: const InputDecoration(  labelText: 'Annual Family Income (in ₹)',  border: OutlineInputBorder(),  ),  style: const TextStyle(color: Colors.*white*),  keyboardType: TextInputType.*number*,  validator: (value) {  if (value == null || value.isEmpty) {  return 'Please enter your family income';  }  if (!RegExp(r'^\d+$').hasMatch(value)) {  return 'Income should contain only numbers';  }  return null;  },  onSaved: (value) => \_formData['familyIncome'] = value!,  ),  const SizedBox(height: 16),  TextFormField(  decoration: const InputDecoration(  labelText: 'Address',  border: OutlineInputBorder(),  ),  style: const TextStyle(color: Colors.*white*),  maxLines: 3,  validator: (value) {  if (value == null || value.isEmpty) {  return 'Please enter your address';  }  if (!RegExp(r'^[a-zA-Z0-9\s,.-]+$').hasMatch(value)) {  return 'Address should contain only letters, numbers, and basic punctuation';  }  return null;  },  onSaved: (value) => \_formData['address'] = value!,  ),  const SizedBox(height: 16),  DropdownButtonFormField<String>(  decoration: const InputDecoration(  labelText: 'District',  border: OutlineInputBorder(),  ),  dropdownColor: const Color(0xFF2D2D2D),  style: const TextStyle(color: Colors.*white*),  items: \_districts  .map((district) => DropdownMenuItem(  value: district,  child: Text(district),  ))  .toList(),  validator: (value) {  if (value == null || value.isEmpty) {  return 'Please select your district';  }  return null;  },  onChanged: (value) => \_formData['district'] = value!,  ),  const SizedBox(height: 16),  TextFormField(  decoration: const InputDecoration(  labelText: 'Aadhar Number',  border: OutlineInputBorder(),  ),  style: const TextStyle(color: Colors.*white*),  keyboardType: TextInputType.*number*,  validator: (value) {  if (value == null || value.isEmpty) {  return 'Please enter your Aadhar number';  }  if (!RegExp(r'^\d{12}$').hasMatch(value)) {  return 'Aadhar number must be 12 digits';  }  return null;  },  onSaved: (value) => \_formData['aadharNumber'] = value!,  ),  const SizedBox(height: 16),  TextFormField(  decoration: const InputDecoration(  labelText: 'Bank Account Number',  border: OutlineInputBorder(),  ),  style: const TextStyle(color: Colors.*white*),  keyboardType: TextInputType.*number*,  validator: (value) {  if (value == null || value.isEmpty) {  return 'Please enter your bank account number';  }  if (!RegExp(r'^\d{9,18}$').hasMatch(value)) {  return 'Bank account number must be between 9 and 18 digits';  }  return null;  },  onSaved: (value) => \_formData['bankAccount'] = value!,  ),  const SizedBox(height: 16),  TextFormField(  decoration: const InputDecoration(  labelText: 'IFSC Code',  border: OutlineInputBorder(),  ),  style: const TextStyle(color: Colors.*white*),  validator: (value) {  if (value == null || value.isEmpty) {  return 'Please enter your IFSC code';  }  if (!\_isValidIFSC(value)) {  return 'IFSC code must be 7 characters (3 letters + 4 numbers)';  }  return null;  },  onSaved: (value) => \_formData['ifscCode'] = value!,  ),  const SizedBox(height: 16),  TextFormField(  decoration: const InputDecoration(  labelText: 'Institution Name',  border: OutlineInputBorder(),  ),  style: const TextStyle(color: Colors.*white*),  validator: (value) {  if (value == null || value.isEmpty) {  return 'Please enter your institution name';  }  return null;  },  onSaved: (value) => \_formData['institutionName'] = value!,  ),  const SizedBox(height: 16),  DropdownButtonFormField<String>(  decoration: const InputDecoration(  labelText: 'Course',  border: OutlineInputBorder(),  ),  dropdownColor: const Color(0xFF2D2D2D),  style: const TextStyle(color: Colors.*white*),  items: \_courses  .map((course) => DropdownMenuItem(  value: course,  child: Text(course),  ))  .toList(),  validator: (value) {  if (value == null || value.isEmpty) {  return 'Please select your course';  }  return null;  },  onChanged: (value) => \_formData['courseName'] = value!,  ),  const SizedBox(height: 32),  ElevatedButton(  onPressed: \_submitForm,  child: const Text(  'Submit Application',  style: TextStyle(  fontSize: 18,  fontWeight: FontWeight.*bold*,  ),  ),  ),  ],  ),  ),  ),  );  } }  Status\_screen.dart  import 'package:flutter/material.dart'; import 'package:provider/provider.dart'; import '../providers/scholarship\_provider.dart';  class StatusScreen extends StatelessWidget {  const StatusScreen({super.key});   String \_getRejectionReason(Map<String, dynamic> data) {  double familyIncome = double.*tryParse*(data['familyIncome'] ?? '0') ?? 0;  String caste = data['caste'] ?? '';   if (caste.toLowerCase() == 'open') {  return 'Scholarship is not available for Open category candidates.';  }  if (familyIncome >= 800000) {  return 'Your family income exceeds the maximum limit of ₹8,00,000 per annum.';  }  if (!caste.toLowerCase().contains('sc') &&   !caste.toLowerCase().contains('st') &&   !caste.toLowerCase().contains('obc')) {  return 'This scholarship is only available for SC/ST/OBC categories.';  }  return 'You do not meet the eligibility criteria for this scholarship.';  }   @override  Widget build(BuildContext context) {  // 60-30-10 color rule  const Color primaryDark = Color(0xFF1A1A1A); // 60% - Main background  const Color secondaryDark = Color(0xFF2D2D2D); // 30% - Secondary background  const Color accentColor = Color(0xFF6C63FF); // 10% - Accent color   return Scaffold(  backgroundColor: primaryDark,  appBar: AppBar(  backgroundColor: secondaryDark,  elevation: 0,  title: const Text(  'Application Status',  style: TextStyle(  fontWeight: FontWeight.*bold*,  color: Colors.*white*,  ),  ),  ),  body: Consumer<ScholarshipProvider>(  builder: (context, provider, child) {  if (!provider.isProcessed) {  return Center(  child: Column(  mainAxisAlignment: MainAxisAlignment.center,  children: [  Icon(  Icons.*description\_outlined*,  size: 64,  color: accentColor.withOpacity(0.5),  ),  const SizedBox(height: 16),  Text(  'No Application Submitted',  style: Theme.*of*(context).textTheme.headlineMedium?.copyWith(  color: Colors.*white*,  ),  ),  const SizedBox(height: 8),  Text(  'Please submit an application first',  style: Theme.*of*(context).textTheme.bodyLarge?.copyWith(  color: Colors.*white70*,  ),  ),  ],  ),  );  }   return SingleChildScrollView(  padding: const EdgeInsets.all(24.0),  child: Column(  crossAxisAlignment: CrossAxisAlignment.stretch,  children: [  Card(  elevation: 8,  color: secondaryDark,  shape: RoundedRectangleBorder(  borderRadius: BorderRadius.circular(20),  ),  child: Padding(  padding: const EdgeInsets.all(24.0),  child: Column(  children: [  Container(  padding: const EdgeInsets.all(16),  decoration: BoxDecoration(  color: provider.isEligible  ? const Color(0xFF4CAF50).withOpacity(0.1)  : const Color(0xFFE53935).withOpacity(0.1),  borderRadius: BorderRadius.circular(100),  ),  child: Icon(  provider.isEligible  ? Icons.*check\_circle\_outline\_rounded* : Icons.*cancel\_outlined*,  size: 48,  color: provider.isEligible  ? const Color(0xFF4CAF50)  : const Color(0xFFE53935),  ),  ),  const SizedBox(height: 24),  Text(  provider.isEligible  ? 'Application Approved!'  : 'Application Not Eligible',  style: Theme.*of*(context).textTheme.headlineMedium?.copyWith(  color: provider.isEligible  ? const Color(0xFF4CAF50)  : const Color(0xFFE53935),  fontWeight: FontWeight.*bold*,  ),  ),  const SizedBox(height: 16),  Text(  provider.isEligible  ? 'Your scholarship amount of ₹1,50,000 will be transferred to your bank account.'  : \_getRejectionReason(provider.applicationData),  textAlign: TextAlign.center,  style: Theme.*of*(context).textTheme.bodyLarge?.copyWith(  color: Colors.*white70*,  ),  ),  if (!provider.isEligible) ...[  const SizedBox(height: 16),  const Divider(color: Colors.*white24*),  const SizedBox(height: 16),  Text(  'Eligibility Criteria:',  style: Theme.*of*(context).textTheme.titleMedium?.copyWith(  fontWeight: FontWeight.*bold*,  color: Colors.*white*,  ),  ),  const SizedBox(height: 8),  \_buildCriteriaItem(context, 'Family income should be below ₹8,00,000 per annum'),  \_buildCriteriaItem(context, 'Must belong to SC/ST/OBC category'),  \_buildCriteriaItem(context, 'Must be a resident of Maharashtra'),  \_buildCriteriaItem(context, 'Must be enrolled in a recognized educational institution'),  ],  ],  ),  ),  ),  const SizedBox(height: 24),  Text(  'Application Details',  style: Theme.*of*(context).textTheme.headlineMedium?.copyWith(  color: Colors.*white*,  fontWeight: FontWeight.*bold*,  ),  ),  const SizedBox(height: 16),  Card(  elevation: 8,  color: secondaryDark,  shape: RoundedRectangleBorder(  borderRadius: BorderRadius.circular(20),  ),  child: Padding(  padding: const EdgeInsets.all(24.0),  child: Column(  crossAxisAlignment: CrossAxisAlignment.start,  children: [  \_buildDetailRow(context, 'Name', provider.applicationData['fullName']),  \_buildDetailRow(context, 'Caste', provider.applicationData['caste']),  \_buildDetailRow(context, 'Family Income', '₹${provider.applicationData['familyIncome']}'),  \_buildDetailRow(context, 'Institution', provider.applicationData['institutionName']),  \_buildDetailRow(context, 'Course', provider.applicationData['courseName']),  ],  ),  ),  ),  ],  ),  );  },  ),  );  }   Widget \_buildDetailRow(BuildContext context, String label, String? value) {  return Padding(  padding: const EdgeInsets.only(bottom: 16.0),  child: Row(  crossAxisAlignment: CrossAxisAlignment.start,  children: [  Expanded(  flex: 2,  child: Text(  label,  style: Theme.*of*(context).textTheme.bodyLarge?.copyWith(  color: Colors.*white70*,  ),  ),  ),  Expanded(  flex: 3,  child: Text(  value ?? 'N/A',  style: Theme.*of*(context).textTheme.bodyLarge?.copyWith(  fontWeight: FontWeight.*w500*,  color: Colors.*white*,  ),  ),  ),  ],  ),  );  }   Widget \_buildCriteriaItem(BuildContext context, String text) {  return Padding(  padding: const EdgeInsets.only(bottom: 8.0),  child: Row(  crossAxisAlignment: CrossAxisAlignment.start,  children: [  Icon(  Icons.*circle*,  size: 8,  color: const Color(0xFF6C63FF),  ),  const SizedBox(width: 8),  Expanded(  child: Text(  text,  style: Theme.*of*(context).textTheme.bodyMedium?.copyWith(  color: Colors.*white70*,  ),  ),  ),  ],  ),  );  } }  scholarship\_provider.dart import 'package:flutter/foundation.dart';  class ScholarshipProvider with ChangeNotifier {  bool \_isEligible = false;  bool \_isProcessed = false;  Map<String, dynamic> \_applicationData = {};   bool get isEligible => \_isEligible;  bool get isProcessed => \_isProcessed;  Map<String, dynamic> get applicationData => \_applicationData;   void submitApplication(Map<String, dynamic> data) {  \_applicationData = data;    // Check eligibility based on income and caste  double familyIncome = double.*tryParse*(data['familyIncome'] ?? '0') ?? 0;  String caste = data['caste'] ?? '';    // Eligibility criteria: Income below 8 lakhs and belongs to reserved category  \_isEligible = familyIncome < 800000 &&   (caste.toLowerCase() != 'open') &&  (caste.toLowerCase().contains('sc') ||   caste.toLowerCase().contains('st') ||   caste.toLowerCase().contains('obc'));    \_isProcessed = true;  notifyListeners();  }   void resetApplication() {  \_isEligible = false;  \_isProcessed = false;  \_applicationData = {};  notifyListeners();  } } |
| **Screenshots** |  |
| **Question and Answers** | 1. **Which Widgets you have used for the application?**  Core Widgets: MaterialApp, Scaffold, AppBar, BottomNavigationBar, Card, Column, Row, SingleChildScrollView, ElevatedButton, TextFormField, DropdownButtonFormField, InkWell, Icon, Consumer (from Provider).  Layout Widgets: Padding, SizedBox, Expanded, Container.  Validation/Input: Form, GlobalKey, DatePicker, SnackBar.  Theming: ThemeData, GoogleFonts.inter, custom color schemes. 2. **Benefits of using flutter**   Cross-Platform: Single codebase for iOS, Android, and web.  Hot Reload: Instant UI updates during development.  Rich Widget Library: Pre-built, customizable widgets for rapid UI development.  Performance: Native ARM code compilation via Dart.  Community & Ecosystem: Extensive packages (e.g., provider, google\_fonts).   1. **For the given theme which colour combination have you used and why?**  Primary (60%): #1A1A1A (dark background) for readability and focus.   Secondary (30%): #2D2D2D (cards, app bars) to create depth and contrast.  Accent (10%): #6C63FF (buttons, icons) for visual hierarchy and CTAs.  Why: Follows the 60-30-10 design rule for balance, ensures accessibility, and aligns with modern dark-mode aesthetics. |
| **Conclusion** | From this session, you learned Flutter app development fundamentals: widget composition (MaterialApp, Scaffold, forms), state management (Provider, ChangeNotifier), UI/UX design (60-30-10 color rule), form validation, navigation, and third-party integrations. You gained hands-on experience building a multi-screen app with themes, user input handling, and business logic implementation for real-world scenarios. |