import 'package:flutter/material.dart';  
import 'package:get/get.dart';  
import 'package:lottie/lottie.dart';  
  
import 'homeScreen.dart';  
  
class SplashScreen extends StatefulWidget {  
 const SplashScreen({super.key});  
  
 @override  
 State<SplashScreen> createState() => \_SplashScreenState();  
}  
  
class \_SplashScreenState extends State<SplashScreen>  
 with SingleTickerProviderStateMixin {  
 late AnimationController \_controller;  
 late Animation<double> \_scaleAnimation;  
  
 @override  
 void initState() {  
 super.initState();  
  
 // Initialize animation controller  
 \_controller = AnimationController(  
 duration: const Duration(seconds: 2),  
 vsync: this,  
 );  
  
 // Scale animation from 0.5x to 1.0x  
 \_scaleAnimation = Tween<double>(begin: 0.5, end: 1.0).animate(  
 CurvedAnimation(parent: \_controller, curve: Curves.*easeInOut*),  
 );  
  
 // Start animation  
 \_controller.forward();  
  
 // Navigate to HomeScreen after animation completes  
 Future.delayed(const Duration(seconds: 2), () {  
 Get.off(() => const HomeScreen());  
 });  
 }  
  
 @override  
 void dispose() {  
 \_controller.dispose();  
 super.dispose();  
 }  
  
 @override  
 Widget build(BuildContext context) {  
 return Scaffold(  
 body: Center(  
 child: ScaleTransition(  
 scale: \_scaleAnimation,  
 child: Center(  
 child: Lottie.*asset*('assets/animation/food\_splash.json', width: 200, height: 200),  
 ),  
 ),  
 ),  
 );  
 }  
}

|  |  |
| --- | --- |
| **Experiment** | 7 |
| **Aim** | Create an app for a restaurant to deliver food at doorstep. |
| **Objective** | * To create app for restaurant food delivery app. * To use either side or tabs menu * To create your own theme for the app. * To create Splash screen for the app. |
| **Name** | Atharva Vasant Angre |
| **UCID** | 2024510001 |
| **Class** | FYMCA |
| **Batch** | A |
| **Date of Submission** | 12.04.2024 |

|  |  |
| --- | --- |
| **Technology used** | **Flutter** |
| **Task** | Create Restaurant App for food delivery at your doorstep. Consider your own creativity and theme. Make use of multiple widgets and themes for make it look the best. |
| **Code with proper label** | Main.dart import 'package:flutter/material.dart'; import 'package:get/get.dart'; import 'package:lab\_7/routes.dart'; import 'package:lab\_7/theme.dart';  import 'Screens/splashScreen.dart'; import 'controller/RestaurantMenuController.dart';  void main() {  Get.put(RestaurantMenuController(), permanent: true);  runApp(const MyApp()); }  class MyApp extends StatelessWidget {  const MyApp({super.key});   @override  Widget build(BuildContext context) {  return GetMaterialApp(  debugShowCheckedModeBanner: false,  theme: AppTheme.*lightTheme*,  home: const SplashScreen(),  getPages: AppRoutes.*routes*,  );  } }  Routes.dart import 'package:get/get.dart'; import 'package:lab\_7/Screens/restaurantDetailScreen.dart'; import 'Screens/cart\_screen.dart'; import 'Screens/homeScreen.dart'; import 'Screens/restaurant\_details.dart';   class AppRoutes {  static final *routes* = [  GetPage(name: '/', page: () => const HomeScreen()),  GetPage(name: '/cart', page: () => CartScreen()),  GetPage(name: '/restaurant', page: () => const RestaurantScreen()),  GetPage(name: '/restaurantDetail', page: () => RestaurantDetailScreen()),  ]; }  themes.dart import 'package:flutter/material.dart';  class AppTheme {  static const Color *primaryColor* = Color(0xFF1565C0); // Blue  static const Color *secondaryColor* = Color(0xFFD32F2F); // Red  static const Color *accentColor* = Color(0xFFFFFFFF); // White   static ThemeData *lightTheme* = ThemeData(  primaryColor: *primaryColor*,  colorScheme: ColorScheme.light(  primary: *primaryColor*,  secondary: *secondaryColor*,  ),  fontFamily: 'Jonesy',  textTheme: const TextTheme(  displayLarge: TextStyle(fontFamily: 'JonesyCapitals', fontSize: 32, fontWeight: FontWeight.*bold*),  displayMedium: TextStyle(fontFamily: 'JonesyCapitals', fontSize: 28, fontWeight: FontWeight.*bold*),  displaySmall: TextStyle(fontFamily: 'JonesyCapitals', fontSize: 24, fontWeight: FontWeight.*bold*),  bodyLarge: TextStyle(fontFamily: 'Jonesy', fontSize: 16),  bodyMedium: TextStyle(fontFamily: 'Jonesy', fontSize: 14),  labelLarge: TextStyle(fontFamily: 'JonesyCapitals', fontSize: 18, fontWeight: FontWeight.*bold*),  ),  appBarTheme: const AppBarTheme(  backgroundColor: *primaryColor*,  foregroundColor: Colors.*white*,  titleTextStyle: TextStyle(fontFamily: 'JonesyCapitals', fontSize: 20, fontWeight: FontWeight.*bold*),  ),  floatingActionButtonTheme: const FloatingActionButtonThemeData(  backgroundColor: *secondaryColor*,  foregroundColor: Colors.*white*,  ),  ); }  restaurant\_controller.dart import 'package:get/get.dart';  class RestaurantMenuController extends GetxController {  // Menu data for the current restaurant  var categories = [  {  "title": "Recommended",  "items": [  {  "name": "Tea",  "price": 30,  "rating": 4.8,  "image": "assets/images/tea.png"  },  ]  },  {  "title": "Snacks",  "items": []  },  {  "title": "Meals - Indian Veg",  "items": [  {  "name": "Paneer Bhurji",  "price": 355,  "rating": 4.5,  "image": "assets/images/panneerbhurji.png"  },  ]  }  ].obs;   // Cart data structure: { "restaurantName": { "itemName": quantity } }  var cart = {}.obs;   void addToCart(String restaurantName, String itemName) {  if (cart.containsKey(restaurantName)) {  Map<String, int> items = cart[restaurantName];  if (items.containsKey(itemName)) {  items[itemName] = items[itemName]! + 1;  } else {  items[itemName] = 1;  }  cart[restaurantName] = items;  } else {  cart[restaurantName] = {itemName: 1};  }  }   void removeFromCart(String restaurantName, String itemName) {  if (cart.containsKey(restaurantName)) {  Map<String, int> items = cart[restaurantName];  if (items.containsKey(itemName) && items[itemName]! > 0) {  items[itemName] = items[itemName]! - 1;  if (items[itemName] == 0) {  items.remove(itemName);  }  }  if (items.isEmpty) {  cart.remove(restaurantName);  } else {  cart[restaurantName] = items;  }  }  }   int getItemCount(String restaurantName, String itemName) {  if (cart.containsKey(restaurantName)) {  Map<String, int> items = cart[restaurantName];  return items[itemName] ?? 0;  }  return 0;  } }  home\_screen.dart import 'package:flutter/material.dart'; import 'package:get/get.dart'; import 'package:lab\_7/Screens/restaurant\_details.dart';  class HomeScreen extends StatelessWidget {  const HomeScreen({super.key});   @override  Widget build(BuildContext context) {  return Scaffold(  appBar: AppBar(  title: const Text('Foodzz'),  ),  body: Center(  child: RestaurantScreen()  ),  floatingActionButton: FloatingActionButton(  onPressed: () {  Get.toNamed('/cart');  },  child: const Icon(Icons.*shopping\_cart*),  ),  );  } }  restaurant\_details.dart import 'package:flutter/material.dart'; import 'package:get/get.dart'; import '../theme.dart';  class RestaurantScreen extends StatelessWidget {  const RestaurantScreen({super.key});   @override  Widget build(BuildContext context) {  List<Map<String, String>> restaurants = [  {  "name": "Pizza Hut",  "rating": "4.3",  "time": "45-50 mins",  "image": "assets/images/pizzahut.png",  "speciality": "Pizzas",   },  {  "name": "Urban Cafe",  "rating": "4.7",  "time": "50-55 mins",  "image": "assets/images/urbancafe.png",  "speciality": "Snacks, Pizzas, Pastas, Fast Food, Burgers, Cafe",   },  {  "name": "Shree Naivedyam",  "rating": "4.5",  "time": "45-50 mins",  "image": "assets/images/shri.png",  "speciality": "North Indian, Chinese, South Indian, Pizzas, Beverages",  },{  "name": "Jai Ganesh Bhojnalaya",  "rating": "4.5",  "time": "35-40 mins",  "image": "assets/images/ganesh.png",  "speciality": "North Indian, South Indian, Indian, Chinese",  },{  "name": "Hotel Sai Nath & Sai Restaurant",  "rating": "4.3",  "time": "35-40 mins",  "image": "assets/images/sai.png",  "speciality": "North Indian, South Indian, Chinese, Beverages, Fast Food, Desserts",  },{  "name": "Bharat Mewad Ice Cream",  "rating": "4.4",  "time": "35-40 mins",  "image": "assets/images/mewad.png",  "speciality": "Ice Cream, Desserts, Beverages",  },{  "name": "Apni Rasoi Family Dhaba",  "rating": "4.2",  "time": "45-50 mins",  "image": "assets/images/rasoi.png",  "speciality": "North Indian, Indian, South Indian, Chinese",  },{  "name": "The Fusion Lounge",  "rating": "4.1",  "time": "50-55 mins",  "image": "assets/images/fusion.png",  "speciality": "South Indian, Chinese, Beverages, Fast Food, Desserts",  },{  "name": "Satkar Restaurant",  "rating": "4.5",  "time": "25-30 mins",  "image": "assets/images/satkar.png",  "speciality": "North Indian, South Indian, Indian, Salads, Desserts",  },  ];   return Scaffold(  body: Padding(  padding: const EdgeInsets.all(8.0),  child: GridView.builder(  gridDelegate: SliverGridDelegateWithFixedCrossAxisCount(  crossAxisCount: 2,  crossAxisSpacing: 10,  mainAxisSpacing: 10,  childAspectRatio: 0.9,  ),  itemCount: restaurants.length,  itemBuilder: (context, index) {  return GestureDetector(  onTap: () {  // Navigate using GetX  Get.toNamed('/restaurantDetail', arguments: restaurants[index]);  },  child: Card(  color: AppTheme.*accentColor*,  elevation: 4,  shape: RoundedRectangleBorder(  borderRadius: BorderRadius.circular(12),  ),  child: Container(  height: Get.height \* 0.3,  child: Column(  crossAxisAlignment: CrossAxisAlignment.start,  children: [  Padding(  padding: const EdgeInsets.all(8.0),  child: ClipRRect(  borderRadius: const BorderRadius.all(Radius.circular(12)),  child: Stack(  children: [  Image.asset(  restaurants[index]['image']!,  width: double.*infinity*,  height: 120,  fit: BoxFit.cover,  ),  Positioned(  bottom: 0,  left: 0,  right: 0,  child: Container(  height: 24,  decoration: BoxDecoration(  gradient: LinearGradient(  begin: Alignment.*bottomCenter*,  end: Alignment.*topCenter*,  colors: [  Colors.*black*.withOpacity(1),  Colors.*transparent*,  ],  ),  ),  ),  ),  ],  ),  ),  ),  Padding(  padding: const EdgeInsets.symmetric(horizontal: 5.0),  child: Column(  crossAxisAlignment: CrossAxisAlignment.start,  children: [  Text(  restaurants[index]['name']!,  overflow: TextOverflow.ellipsis,  style: const TextStyle(  fontSize: 20,  fontWeight: FontWeight.*bold*,  ),  ),  Row(  children: [  const Icon(Icons.*stars*, color: Colors.*green*),  Text(  '${restaurants[index]['rating']} - ${restaurants[index]['time']}',  style: TextStyle(color: Colors.*grey*[600]),  ),  ],  ),  Text(  restaurants[index]['speciality']!,  style: TextStyle(  fontSize: 16,  fontWeight: FontWeight.*w400*,  color: Colors.*grey*[600],  ),  overflow: TextOverflow.ellipsis,  ),  ],  ),  ),  ],  ),  ),  ),  );  },  ),  ),  );  } }  restaurant\_details\_screen.dart import 'package:flutter/material.dart'; import 'package:get/get.dart'; import '../controller/RestaurantMenuController.dart';  class RestaurantDetailScreen extends StatelessWidget {  // Retrieve the global controller instance.  final RestaurantMenuController menuController = Get.find<RestaurantMenuController>();   @override  Widget build(BuildContext context) {  final Map<String, dynamic> restaurant =  Get.arguments?.cast<String, dynamic>() ?? {};  final String restaurantName = restaurant['name'] ?? 'Restaurant Name';   return Scaffold(  appBar: AppBar(title: const Text("Restaurant Menu")),  body: Padding(  padding: const EdgeInsets.all(8.0),  child: SingleChildScrollView(  child: Column(  crossAxisAlignment: CrossAxisAlignment.start,  children: [  // Restaurant Image and Gradient overlay  ClipRRect(  borderRadius: const BorderRadius.all(Radius.circular(12)),  child: Stack(  children: [  Image.asset(  restaurant['image']!,  width: double.*infinity*,  height: 200,  fit: BoxFit.cover,  ),  Positioned(  bottom: 0,  left: 0,  right: 0,  child: Container(  height: 24,  decoration: BoxDecoration(  gradient: LinearGradient(  begin: Alignment.*bottomCenter*,  end: Alignment.*topCenter*,  colors: [  Colors.*black*.withOpacity(1),  Colors.*transparent*,  ],  ),  ),  ),  ),  ],  ),  ),  const SizedBox(height: 16),  // Restaurant Details  Padding(  padding: const EdgeInsets.symmetric(horizontal: 16.0),  child: Column(  crossAxisAlignment: CrossAxisAlignment.start,  children: [  Text(  restaurantName,  style: const TextStyle(  fontSize: 24, fontWeight: FontWeight.*bold*),  ),  const SizedBox(height: 8),  Row(  children: [  const Icon(Icons.*star*, color: Colors.*green*),  Text(  restaurant['rating'] ?? 'N/A',  style: const TextStyle(  fontSize: 18, color: Colors.*black87*),  ),  ],  ),  const SizedBox(height: 8),  Text(  "Estimated Delivery Time: ${restaurant['time'] ?? 'N/A'}",  style: const TextStyle(  fontSize: 16, color: Colors.*black54*),  ),  const SizedBox(height: 8),  Text(  "Specialities: ${restaurant['speciality'] ?? 'N/A'}",  style: const TextStyle(  fontSize: 16, color: Colors.*black54*),  ),  ],  ),  ),  const SizedBox(height: 10),  Center(  child: Text(  "Menu",  style: TextStyle(fontSize: 25),  ),  ),  // Menu List  Obx(() {  return ListView.builder(  shrinkWrap: true,  physics: const NeverScrollableScrollPhysics(),  itemCount: menuController.categories.length,  itemBuilder: (context, index) {  var category = menuController.categories[index];  var items = (category['items'] as List?) ?? [];  return ExpansionTile(  title: Text(  "${category['title']} (${items.length})",  style: const TextStyle(fontWeight: FontWeight.*bold*),  ),  children: items.map<Widget>((item) {  return Stack(  children: [  Container(  padding: const EdgeInsets.symmetric(  vertical: 15, horizontal: 16),  child: Row(  mainAxisAlignment:  MainAxisAlignment.spaceBetween,  children: [  // Icon (if needed)  Icon(Icons.*indeterminate\_check\_box\_outlined*,  color: Colors.*green*),  // Food Name and Price  Expanded(  child: Column(  crossAxisAlignment:  CrossAxisAlignment.start,  children: [  Text(  item["name"],  style: const TextStyle(  fontWeight: FontWeight.*bold*,  fontSize: 16),  ),  Text(  "₹${item["price"]}",  style: const TextStyle(  fontWeight: FontWeight.*bold*,  color: Colors.*black87*),  ),  ],  ),  ),  // Food Image  ClipRRect(  borderRadius: BorderRadius.circular(8),  child: Image.asset(  item["image"],  width: 100,  height: 100,  fit: BoxFit.cover,  ),  ),  ],  ),  ),  // Add to Cart UI (always shows "-" and "+" buttons)  Positioned(  top: Get.height \* 0.090,  left: Get.width \* 0.730,  child: Obx(() {  int itemCount = menuController.getItemCount(  restaurantName, item["name"]);  return Container(  padding: const EdgeInsets.all(8.0),  decoration: BoxDecoration(  borderRadius: BorderRadius.circular(10),  color: Colors.*green*,  ),  child: Row(  mainAxisAlignment:  MainAxisAlignment.spaceEvenly,  children: [  GestureDetector(  onTap: () {  if (itemCount > 0) {  menuController.removeFromCart(  restaurantName, item["name"]);  Get.snackbar("Item Removed Successfully",  "",  snackPosition:  SnackPosition.BOTTOM,  backgroundColor: Colors.*red*);  }  },  child: const Text(  "-",  style: TextStyle(color: Colors.*white*),  ),  ),  Padding(  padding: const EdgeInsets.symmetric(  horizontal: 8.0),  child: Text(  itemCount == 0 ? "Add" : " $itemCount ",  style: const TextStyle(  color: Colors.*white*),  ),  ),  GestureDetector(  onTap: () {  menuController.addToCart(  restaurantName, item["name"]);  Get.snackbar("Item Added Successfully", "",  snackPosition: SnackPosition.BOTTOM,  backgroundColor: Colors.*green*);  },  child: const Text(  "+",  style: TextStyle(color: Colors.*white*),  ),  ),  ],  ),  );  }),  ),  ],  );  }).toList(),  );  },  );  }),  ],  ),  ),  ),  floatingActionButton: FloatingActionButton(  onPressed: () {  Get.toNamed('/cart');  },  child: const Icon(Icons.*shopping\_cart*),  ),  );  } }  cart\_Screen.dart import 'package:flutter/material.dart'; import 'package:get/get.dart'; import '../controller/RestaurantMenuController.dart';  class CartScreen extends StatelessWidget {  final RestaurantMenuController menuController = Get.find<RestaurantMenuController>();   final double deliveryFee = 71.0;  final double taxRate = 0.08;   CartScreen({Key? key}) : super(key: key);   @override  Widget build(BuildContext context) {  return Scaffold(  appBar: AppBar(  title: const Text('Cart'),  ),  body: Obx(() {  if (menuController.cart.isEmpty) {  return const Center(  child: Text('Your cart is empty'),  );  }   double itemTotal = 0.0;  menuController.cart.forEach((restaurantName, itemsMap) {  (itemsMap as Map<String, int>).forEach((itemName, quantity) {  var item = menuController.categories  .expand((category) => category["items"] as List)  .firstWhere(  (element) => element["name"] == itemName,  orElse: () => null,  );  if (item != null) {  itemTotal += (item["price"] \* quantity);  }  });  });   double taxAndOtherCharges = itemTotal \* taxRate;   double tip = 0.0;   double toPay = itemTotal + deliveryFee + taxAndOtherCharges + tip;   return Column(  children: [  Expanded(  child: ListView(  children: menuController.cart.entries.map<Widget>((restaurantEntry) {  final restaurantName = restaurantEntry.key;  final itemsMap = restaurantEntry.value as Map<String, int>;   // Build a list of Widgets for each restaurant  List<Widget> listItems = [];   // Restaurant header  listItems.add(  Padding(  padding: const EdgeInsets.all(8.0),  child: Text(  restaurantName,  style: const TextStyle(  fontSize: 20,  fontWeight: FontWeight.*bold*,  ),  ),  ),  );   // Each item in the restaurant  itemsMap.entries.forEach((entry) {  String itemName = entry.key;  int quantity = entry.value;   var item = menuController.categories  .expand((category) => category["items"] as List)  .firstWhere(  (element) => element["name"] == itemName,  orElse: () => null,  );   listItems.add(  ListTile(  leading: item != null  ? Image.asset(item["image"], width: 50, height: 50)  : null,  title: Text(itemName),  trailing: Row(  mainAxisSize: MainAxisSize.min,  children: [  // Quantity control  Container(  decoration: BoxDecoration(  borderRadius: BorderRadius.circular(50),  border: Border.all(color: Colors.*black*),  ),  child: Row(  children: [  IconButton(  onPressed: () => menuController.removeFromCart(  restaurantName,  itemName,  ),  icon: const Icon(Icons.*remove*),  ),  Text("$quantity"),  IconButton(  onPressed: () => menuController.addToCart(  restaurantName,  itemName,  ),  icon: const Icon(Icons.*add*),  ),  ],  ),  ),  Padding(  padding: const EdgeInsets.only(left: 20),  child: Text(  "₹${quantity \* (item?["price"] ?? 0)}",  ),  ),  ],  ),  ),  );  });   return Column(  crossAxisAlignment: CrossAxisAlignment.start,  children: listItems,  );  }).toList(),  ),  ),   Container(  width: double.*infinity*,  padding: const EdgeInsets.symmetric(horizontal: 16, vertical: 12),  decoration: const BoxDecoration(  color: Colors.*white*,  border: Border(  top: BorderSide(color: Colors.*black12*),  ),  ),  child: Column(  crossAxisAlignment: CrossAxisAlignment.start,  children: [  const Text(  "Bill Details",  style: TextStyle(fontWeight: FontWeight.*bold*, fontSize: 16),  ),  const SizedBox(height: 8),   Row(  mainAxisAlignment: MainAxisAlignment.spaceBetween,  children: [  const Text("Item Total"),  Text("₹${itemTotal.toStringAsFixed(2)}"),  ],  ),   Row(  mainAxisAlignment: MainAxisAlignment.spaceBetween,  children: [  const Text("Delivery Fee"),  Text("₹${deliveryFee.toStringAsFixed(2)}"),  ],  ),  const SizedBox(height: 8),  const Divider(thickness: 1),   Row(  mainAxisAlignment: MainAxisAlignment.spaceBetween,  children: [  const Text("Delivery Tip"),  GestureDetector(  onTap: () {   },  child: const Text(  "Add tip",  style: TextStyle(  color: Colors.*red*,  ),  ),  ),  ],  ),   Row(  mainAxisAlignment: MainAxisAlignment.spaceBetween,  children: [  const Text("GST & Other Charges"),  Text("₹${taxAndOtherCharges.toStringAsFixed(2)}"),  ],  ),   const SizedBox(height: 8),  const Divider(thickness: 1),   Row(  mainAxisAlignment: MainAxisAlignment.spaceBetween,  children: [  const Text(  "TO PAY",  style: TextStyle(  fontWeight: FontWeight.*bold*,  fontSize: 16,  ),  ),  Text(  "₹${toPay.toStringAsFixed(2)}",  style: const TextStyle(  fontWeight: FontWeight.*bold*,  fontSize: 16,  ),  ),  ],  ),  ],  ),  ),  ],  );  }),  );  } }  splash\_screen.dart import 'package:flutter/material.dart'; import 'package:get/get.dart'; import 'package:lottie/lottie.dart';  import 'homeScreen.dart';  class SplashScreen extends StatefulWidget {  const SplashScreen({super.key});   @override  State<SplashScreen> createState() => \_SplashScreenState(); }  class \_SplashScreenState extends State<SplashScreen>  with SingleTickerProviderStateMixin {  late AnimationController \_controller;  late Animation<double> \_scaleAnimation;   @override  void initState() {  super.initState();   // Initialize animation controller  \_controller = AnimationController(  duration: const Duration(seconds: 2),  vsync: this,  );   // Scale animation from 0.5x to 1.0x  \_scaleAnimation = Tween<double>(begin: 0.5, end: 1.0).animate(  CurvedAnimation(parent: \_controller, curve: Curves.*easeInOut*),  );   // Start animation  \_controller.forward();   // Navigate to HomeScreen after animation completes  Future.delayed(const Duration(seconds: 2), () {  Get.off(() => const HomeScreen());  });  }   @override  void dispose() {  \_controller.dispose();  super.dispose();  }   @override  Widget build(BuildContext context) {  return Scaffold(  body: Center(  child: ScaleTransition(  scale: \_scaleAnimation,  child: Center(  child: Lottie.*asset*('assets/animation/food\_splash.json', width: 200, height: 200),  ),  ),  ),  );  } } |
| **Screenshots** |  |
| **Question and Answers** | Answer the following Questions:   1. How did you use 60-30-10 rule in your application?   Used 60% for background, 30% for UI elements like cards, and 10% for buttons or highlights to keep the design balanced.   1. Which new elements did you use for creating UI components?   Used custom widgets, animations, bottom sheets, carousel sliders, and pull-to-refresh for better UI and user experience.   1. In pubspec.yaml file, what dependencies need to be there?   get: ^4.7.2 (State management) lottie: ^3.3.1(To use Lottie files)   1. What is the use of Splash Screen?   It shows the logo or app name while the app is loading. Helps with branding and gives a smooth start. |
| **Conclusion** | Learned to use GetX for state management and navigation easily. Used Lottie to add animated illustrations and make the UI more engaging. Understood how to manage UI, navigation, and animations in a clean and efficient way. Also understood local data management. |