|  |  |
| --- | --- |
| **Experiment** | LAB ESE Submission |
| **Aim** | Create an app for booking Doctors Appointment in a hospital consisting of the following field (Name of the Patient, Address, Age, Contact Details, Symptoms, Name of the doctor (Dropdown), Date of visit (Date picker). The Name, Address, Age, Contact must be saved in firebase. |
| **Name** | Atharva Vasant Angre |
| **UCID** | 2024510001 |
| **Class** | FYMCA |
| **Batch** | A |
| **Date of Examination** | 13.05.2025 |

|  |  |
| --- | --- |
| **Technology used** | **Flutter, Firebase** |
| **Task** | * Make UI to Take user input also properly validate them. * Connect to Firebase. * Make Firebase service to send and receive data. * Make Controller to handle data. |
| **Code with proper label** | **main.dart**  import 'package:flutter/material.dart'; import 'package:firebase\_core/firebase\_core.dart'; import 'package:get/get.dart'; import 'package:lab\_ese/constants/themeConstants.dart'; import 'package:lab\_ese/controllers/appointment\_controller.dart'; import 'package:lab\_ese/screens/home\_screen.dart'; import 'firebase\_options.dart';  Future<void> main() async {  WidgetsFlutterBinding.*ensureInitialized*();  await Firebase.*initializeApp*(  options: DefaultFirebaseOptions.*currentPlatform*,  );  Get.put(AppointmentController());    runApp(const MyApp()); }  class MyApp extends StatelessWidget {  const MyApp({super.key});   // This widget is the root of your application.  @override  Widget build(BuildContext context) {  return GetMaterialApp(  title: 'Doctor Appointment App',  debugShowCheckedModeBanner: false,  theme: ThemeData(  colorScheme: ColorScheme.fromSeed(seedColor: ThemeConstants.*primaryColor*),  useMaterial3: true,  scaffoldBackgroundColor: ThemeConstants.*backgroundColor*,  appBarTheme: const AppBarTheme(  backgroundColor: ThemeConstants.*primaryColor*,  foregroundColor: Colors.*white*,  centerTitle: true,  elevation: 0,  ),  elevatedButtonTheme: ElevatedButtonThemeData(  style: ThemeConstants.*primaryButtonStyle*,  ),  outlinedButtonTheme: OutlinedButtonThemeData(  style: ThemeConstants.*secondaryButtonStyle*,  ),  cardTheme: CardTheme(  elevation: 0,  shape: RoundedRectangleBorder(  borderRadius: BorderRadius.circular(12),  ),  color: ThemeConstants.*cardColor*,  ),  textTheme: const TextTheme(  headlineLarge: ThemeConstants.*headingStyle*,  headlineMedium: ThemeConstants.*subheadingStyle*,  bodyLarge: ThemeConstants.*bodyTextStyle*,  bodyMedium: ThemeConstants.*bodyTextStyle*,  bodySmall: ThemeConstants.*captionStyle*,  ),  ),  home: const HomeScreen(),  );  } }  **Homescreen.dart** import 'package:flutter/material.dart'; import 'package:get/get.dart'; import 'package:lab\_ese/constants/themeConstants.dart'; import 'package:lab\_ese/screens/appointment\_screen.dart'; import 'package:lab\_ese/screens/appointments\_list\_screen.dart';  class HomeScreen extends StatelessWidget {  const HomeScreen({super.key});   @override  Widget build(BuildContext context) {  return Scaffold(  backgroundColor: ThemeConstants.*backgroundColor*,  body: SafeArea(  child: Column(  children: [  Container(  width: double.*infinity*,  decoration: const BoxDecoration(  color: ThemeConstants.*primaryColor*,  borderRadius: BorderRadius.only(  bottomLeft: Radius.circular(30),  bottomRight: Radius.circular(30),  ),  ),  padding: const EdgeInsets.fromLTRB(20, 20, 20, 40),  child: Column(  crossAxisAlignment: CrossAxisAlignment.start,  children: [  const Text(  'Welcome to',  style: TextStyle(  color: Colors.*white70*,  fontSize: 16,  ),  ),  const SizedBox(height: 8),  const Text(  'NeuraLife',  style: TextStyle(  color: Colors.*white*,  fontSize: 24,  fontWeight: FontWeight.*bold*,  ),  ),  const SizedBox(height: 10),  Text(  'Book your doctor appointments easily',  style: TextStyle(  color: Colors.*white*.withOpacity(0.9),  fontSize: 16,  ),  ),  ],  ),  ),    Expanded(  child: SingleChildScrollView(  padding: const EdgeInsets.all(20),  child: Column(  crossAxisAlignment: CrossAxisAlignment.start,  children: [  const Text(  'What would you like to do?',  style: ThemeConstants.*subheadingStyle*,  ),  const SizedBox(height: 20),    InkWell(  onTap: () => Get.to(() => AppointmentScreen()),  child: Container(  decoration: ThemeConstants.*cardDecoration*,  padding: const EdgeInsets.all(20),  margin: const EdgeInsets.only(bottom: 16),  child: Row(  children: [  Container(  decoration: BoxDecoration(  color: ThemeConstants.*primaryColor*.withOpacity(0.1),  borderRadius: BorderRadius.circular(12),  ),  padding: const EdgeInsets.all(12),  child: const Icon(  Icons.*calendar\_month\_rounded*,  color: ThemeConstants.*primaryColor*,  size: 32,  ),  ),  const SizedBox(width: 20),  Expanded(  child: Column(  crossAxisAlignment: CrossAxisAlignment.start,  children: [  const Text(  'Book Appointment',  style: TextStyle(  fontSize: 18,  fontWeight: FontWeight.*bold*,  color: ThemeConstants.*textPrimaryColor*,  ),  ),  const SizedBox(height: 4),  Text(  'Schedule a new appointment with one of our doctors',  style: TextStyle(  color: ThemeConstants.*textSecondaryColor*.withOpacity(0.8),  ),  ),  ],  ),  ),  const Icon(  Icons.*arrow\_forward\_ios\_rounded*,  color: ThemeConstants.*primaryColor*,  size: 16,  ),  ],  ),  ),  ),    InkWell(  onTap: () => Get.to(() => AppointmentsListScreen()),  child: Container(  decoration: ThemeConstants.*cardDecoration*,  padding: const EdgeInsets.all(20),  margin: const EdgeInsets.only(bottom: 16),  child: Row(  children: [  Container(  decoration: BoxDecoration(  color: ThemeConstants.*secondaryColor*.withOpacity(0.1),  borderRadius: BorderRadius.circular(12),  ),  padding: const EdgeInsets.all(12),  child: const Icon(  Icons.*list\_alt\_rounded*,  color: ThemeConstants.*secondaryColor*,  size: 32,  ),  ),  const SizedBox(width: 20),  Expanded(  child: Column(  crossAxisAlignment: CrossAxisAlignment.start,  children: [  const Text(  'View Appointments',  style: TextStyle(  fontSize: 18,  fontWeight: FontWeight.*bold*,  color: ThemeConstants.*textPrimaryColor*,  ),  ),  const SizedBox(height: 4),  Text(  'Check your upcoming appointments',  style: TextStyle(  color: ThemeConstants.*textSecondaryColor*.withOpacity(0.8),  ),  ),  ],  ),  ),  const Icon(  Icons.*arrow\_forward\_ios\_rounded*,  color: ThemeConstants.*secondaryColor*,  size: 16,  ),  ],  ),  ),  ),  ],  ),  ),  ),  ],  ),  ),  );  }    Widget \_buildInfoItem(IconData icon, String text) {  return Column(  children: [  Icon(  icon,  color: ThemeConstants.*primaryColor*,  size: 20,  ),  const SizedBox(height: 4),  Text(  text,  style: const TextStyle(  fontSize: 14,  color: ThemeConstants.*textSecondaryColor*,  ),  ),  ],  );  } }  **Appointmentscreen.dart**  import 'package:flutter/material.dart'; import 'package:get/get.dart'; import 'package:intl/intl.dart'; import 'package:lab\_ese/constants/doctors\_list.dart'; import 'package:lab\_ese/constants/themeConstants.dart'; import 'package:lab\_ese/controllers/appointment\_controller.dart';  class AppointmentScreen extends StatefulWidget {  const AppointmentScreen({super.key});   @override  State<AppointmentScreen> createState() => \_AppointmentScreenState(); }  class \_AppointmentScreenState extends State<AppointmentScreen> {  final GlobalKey<FormState> \_formKey = GlobalKey<FormState>();  final TextEditingController \_nameController = TextEditingController();  final TextEditingController \_addressController = TextEditingController();  final TextEditingController \_ageController = TextEditingController();  final TextEditingController \_contactController = TextEditingController();  final TextEditingController \_symptomsController = TextEditingController();    final AppointmentController \_controller = Get.find<AppointmentController>();   Future<void> \_selectDate(BuildContext context) async {  final DateTime? picked = await showDatePicker(  context: context,  initialDate: \_controller.appointmentDate.value,  firstDate: DateTime.now(),  lastDate: DateTime.now().add(const Duration(days: 365)),  builder: (context, child) {  return Theme(  data: Theme.*of*(context).copyWith(  colorScheme: const ColorScheme.light(  primary: ThemeConstants.*primaryColor*,  onPrimary: Colors.*white*,  onSurface: ThemeConstants.*textPrimaryColor*,  ),  ),  child: child!,  );  },  );  if (picked != null) {  \_controller.setAppointmentDate(picked);  }  }   Future<void> \_saveAppointment() async {  if (\_formKey.currentState!.validate()) {  // Transfer values from controllers to GetX controller  \_controller.name.value = \_nameController.text;  \_controller.address.value = \_addressController.text;  \_controller.age.value = \_ageController.text;  \_controller.contactDetails.value = \_contactController.text;  \_controller.symptoms.value = \_symptomsController.text;    final result = await \_controller.saveAppointment();    if (result) {  Get.back();  Get.snackbar(  'Success',  'Appointment booked successfully',  snackPosition: SnackPosition.BOTTOM,  backgroundColor: ThemeConstants.*successColor*,  colorText: Colors.*white*,  margin: const EdgeInsets.all(10),  borderRadius: 10,  );  } else {  Get.snackbar(  'Error',  'Failed to book appointment',  snackPosition: SnackPosition.BOTTOM,  backgroundColor: ThemeConstants.*errorColor*,  colorText: Colors.*white*,  margin: const EdgeInsets.all(10),  borderRadius: 10,  );  }  }  }   @override  void dispose() {  \_nameController.dispose();  \_addressController.dispose();  \_ageController.dispose();  \_contactController.dispose();  \_symptomsController.dispose();  super.dispose();  }   @override  Widget build(BuildContext context) {  return Scaffold(  backgroundColor: ThemeConstants.*backgroundColor*,  appBar: AppBar(  title: const Text('Book Appointment'),  centerTitle: true,  ),  body: Obx(() => \_controller.isLoading.value  ? const Center(child: CircularProgressIndicator())  : SingleChildScrollView(  padding: const EdgeInsets.all(20.0),  child: Form(  key: \_formKey,  child: Column(  crossAxisAlignment: CrossAxisAlignment.start,  children: [  const SizedBox(height: 10),   Container(  width: double.*infinity*,  padding: const EdgeInsets.all(20),  decoration: BoxDecoration(  color: ThemeConstants.*primaryColor*,  borderRadius: BorderRadius.circular(12),  ),  child: Column(  crossAxisAlignment: CrossAxisAlignment.start,  children: [  const Text(  'Complete Your Appointment Details',  style: TextStyle(  color: Colors.*white*,  fontSize: 18,  fontWeight: FontWeight.*bold*,  ),  ),  const SizedBox(height: 8),  Text(  'Fill in the form below to book your appointment with our specialists',  style: TextStyle(  color: Colors.*white*.withOpacity(0.9),  fontSize: 14,  ),  ),  ],  ),  ),  const SizedBox(height: 20),  const Text(  'Patient Information',  style: ThemeConstants.*subheadingStyle*,  ),  const SizedBox(height: 16),  \_buildFormField(  controller: \_nameController,  label: 'Patient Name',  icon: Icons.*person*,  validator: (value) {  if (value == null || value.isEmpty || value.length <= 2) {  return 'Please enter patient name';  }  return null;  },  ),  const SizedBox(height: 16),  \_buildFormField(  controller: \_addressController,  label: 'Address',  icon: Icons.*home*,  validator: (value) {  if (value == null || value.isEmpty) {  return 'Please enter address';  }  return null;  },  ),  const SizedBox(height: 16),  Row(  children: [  Expanded(  child: \_buildFormField(  controller: \_ageController,  label: 'Age',  icon: Icons.*calendar\_today*,  keyboardType: TextInputType.*number*,  validator: (value) {  if (value == null || value.isEmpty) {  return 'Please enter age';  }  if (int.*tryParse*(value) == null) {  return 'Please enter a valid number';  }  return null;  },  ),  ),  const SizedBox(width: 16),  Expanded(  child: \_buildFormField(  controller: \_contactController,  label: 'Contact Details',  icon: Icons.*phone*,  keyboardType: TextInputType.*phone*,  validator: (value) {  if (value == null || value.isEmpty) {  return 'Please enter contact details';  }  if (value.length != 10) {  return 'Please enter proper phone number';  }  return null;  },  ),  ),  ],  ),  const SizedBox(height: 16),  \_buildFormField(  controller: \_symptomsController,  label: 'Symptoms',  icon: Icons.*medical\_services*,  maxLines: 3,  validator: (value) {  if (value == null || value.isEmpty) {  return 'Please enter symptoms';  }  return null;  },  ),  const SizedBox(height: 24),  const Text(  'Appointment Details',  style: ThemeConstants.*subheadingStyle*,  ),  const SizedBox(height: 16),  Container(  decoration: BoxDecoration(  color: Colors.*white*,  borderRadius: BorderRadius.circular(8),  border: Border.all(color: ThemeConstants.*textLightColor*),  ),  child: DropdownButtonFormField<String>(  decoration: const InputDecoration(  labelText: 'Select Doctor',  prefixIcon: Icon(Icons.*person\_outline*),  border: InputBorder.*none*,  contentPadding: EdgeInsets.symmetric(horizontal: 16,vertical: 10),  ),  value: \_controller.selectedDoctor.value,  items: doctors.map((Doctor doctor) {  return DropdownMenuItem<String>(  value: doctor.name,  child: Text('${doctor.name} (${doctor.specialization})'),  );  }).toList(),  onChanged: (String? newValue) {  if (newValue != null) {  \_controller.setSelectedDoctor(newValue);  }  },  ),  ),  const SizedBox(height: 16),  InkWell(  onTap: () => \_selectDate(context),  child: Container(  padding: const EdgeInsets.symmetric(horizontal: 16, vertical: 14),  decoration: BoxDecoration(  color: Colors.*white*,  borderRadius: BorderRadius.circular(8),  border: Border.all(color: ThemeConstants.*textLightColor*),  ),  child: Row(  children: [  const Icon(  Icons.*event*,  color: ThemeConstants.*textSecondaryColor*,  ),  const SizedBox(width: 10),  Column(  crossAxisAlignment: CrossAxisAlignment.start,  children: [  const Text(  'Appointment Date',  style: TextStyle(  color: ThemeConstants.*textSecondaryColor*,  fontSize: 12,  ),  ),  const SizedBox(height: 4),  Obx(() => Text(  DateFormat('EEEE, MMM dd, yyyy').format(\_controller.appointmentDate.value),  style: const TextStyle(  fontSize: 16,  color: ThemeConstants.*textPrimaryColor*,  fontWeight: FontWeight.*w500*,  ),  )),  ],  ),  ],  ),  ),  ),  const SizedBox(height: 32),  SizedBox(  width: double.*infinity*,  height: 50,  child: ElevatedButton(  onPressed: \_saveAppointment,  child: const Text(  'Book Appointment',  style: TextStyle(fontSize: 18),  ),  ),  ),  const SizedBox(height: 20),  ],  ),  ),  ),  ),  );  }    Widget \_buildFormField({  required TextEditingController controller,  required String label,  required IconData icon,  int maxLines = 1,  TextInputType keyboardType = TextInputType.*text*,  required String? Function(String?) validator,  }) {  return Container(  decoration: BoxDecoration(  color: Colors.*white*,  borderRadius: BorderRadius.circular(8),  ),  child: TextFormField(  controller: controller,  decoration: InputDecoration(  labelText: label,  prefixIcon: Icon(icon),  border: OutlineInputBorder(  borderRadius: BorderRadius.circular(8),  ),  ),  maxLines: maxLines,  keyboardType: keyboardType,  validator: validator,  ),  );  } }  **Appointmentlist.dart**  import 'package:flutter/material.dart'; import 'package:get/get.dart'; import 'package:intl/intl.dart'; import 'package:lab\_ese/constants/themeConstants.dart'; import 'package:lab\_ese/controllers/appointment\_controller.dart'; import 'package:lab\_ese/models/patient\_model.dart';  class AppointmentsListScreen extends StatelessWidget {  const AppointmentsListScreen({super.key});   @override  Widget build(BuildContext context) {  final AppointmentController controller = Get.find<AppointmentController>();    controller.loadPatients();    return Scaffold(  backgroundColor: ThemeConstants.*backgroundColor*,  appBar: AppBar(  title: const Text('All Appointments'),  actions: [  IconButton(  icon: const Icon(Icons.*refresh*),  onPressed: controller.loadPatients,  ),  ],  ),  body: Obx(() {  if (controller.isLoading.value) {  return const Center(child: CircularProgressIndicator());  }    if (controller.patients.isEmpty) {  return Center(  child: Column(  mainAxisAlignment: MainAxisAlignment.center,  children: [  Icon(  Icons.*calendar\_today\_outlined*,  size: 70,  color: ThemeConstants.*textLightColor*.withOpacity(0.5),  ),  const SizedBox(height: 16),  const Text(  'No appointments found',  style: TextStyle(  fontSize: 18,  color: ThemeConstants.*textSecondaryColor*,  fontWeight: FontWeight.*w600*,  ),  ),  const SizedBox(height: 8),  const Text(  'Book your first appointment now!',  style: TextStyle(  color: ThemeConstants.*textLightColor*,  ),  ),  const SizedBox(height: 24),  ElevatedButton.icon(  onPressed: () {  Get.back();  },  icon: const Icon(Icons.*add*),  label: const Text('Book an Appointment'),  ),  ],  ),  );  }    return ListView.builder(  itemCount: controller.patients.length,  padding: const EdgeInsets.all(16),  itemBuilder: (context, index) {  final patient = controller.patients[index];  return \_buildAppointmentCard(patient);  },  );  }),  );  }    Widget \_buildAppointmentCard(Patient patient) {  final bool isUpcoming = patient.appointmentDate.isAfter(DateTime.now());    return Container(  margin: const EdgeInsets.only(bottom: 16),  decoration: ThemeConstants.*cardDecoration*,  child: Column(  children: [   Container(  padding: const EdgeInsets.symmetric(vertical: 12, horizontal: 16),  decoration: BoxDecoration(  color: isUpcoming ? ThemeConstants.*primaryColor*.withOpacity(0.1) : ThemeConstants.*textLightColor*.withOpacity(0.1),  borderRadius: const BorderRadius.only(  topLeft: Radius.circular(12),  topRight: Radius.circular(12),  ),  ),  child: Row(  children: [  Icon(  Icons.*event*,  size: 20,  color: isUpcoming ? ThemeConstants.*primaryColor* : ThemeConstants.*textSecondaryColor*,  ),  const SizedBox(width: 8),  Text(  DateFormat('EEEE, MMMM dd, yyyy').format(patient.appointmentDate),  style: TextStyle(  fontWeight: FontWeight.*w600*,  color: isUpcoming ? ThemeConstants.*primaryColor* : ThemeConstants.*textSecondaryColor*,  ),  ),  const Spacer(),  Container(  padding: const EdgeInsets.symmetric(horizontal: 10, vertical: 4),  decoration: BoxDecoration(  color: isUpcoming ? ThemeConstants.*primaryColor* : ThemeConstants.*textLightColor*,  borderRadius: BorderRadius.circular(20),  ),  child: Text(  isUpcoming ? 'Upcoming' : 'Past',  style: const TextStyle(  color: Colors.*white*,  fontSize: 12,  fontWeight: FontWeight.*w500*,  ),  ),  ),  ],  ),  ),    // Patient details  Padding(  padding: const EdgeInsets.all(16),  child: Column(  crossAxisAlignment: CrossAxisAlignment.start,  children: [  Row(  crossAxisAlignment: CrossAxisAlignment.center,  children: [  Container(  width: 50,  height: 50,  decoration: BoxDecoration(  color: ThemeConstants.*primaryColor*.withOpacity(0.1),  shape: BoxShape.circle,  ),  child: Center(  child: Text(  patient.name.isNotEmpty ? patient.name[0].toUpperCase() : '?',  style: const TextStyle(  fontSize: 24,  fontWeight: FontWeight.*bold*,  color: ThemeConstants.*primaryColor*,  ),  ),  ),  ),  const SizedBox(width: 16),  Expanded(  child: Column(  crossAxisAlignment: CrossAxisAlignment.start,  children: [  Text(  patient.name,  style: const TextStyle(  fontSize: 18,  fontWeight: FontWeight.*bold*,  color: ThemeConstants.*textPrimaryColor*,  ),  ),  Text(  'Age: ${patient.age} years',  style: const TextStyle(  color: ThemeConstants.*textSecondaryColor*,  ),  ),  ],  ),  ),  ],  ),  const SizedBox(height: 16),  const Divider(),  const SizedBox(height: 8),    // Contact and address  Row(  children: [  Expanded(  child: \_buildInfoItem(  icon: Icons.*phone*,  title: 'Contact',  value: patient.contactDetails,  ),  ),  Expanded(  child: \_buildInfoItem(  icon: Icons.*location\_on\_outlined*,  title: 'Address',  value: patient.address,  ),  ),  ],  ),  const SizedBox(height: 16),    // Doctor and symptoms  Row(  children: [  Expanded(  child: \_buildInfoItem(  icon: Icons.*medical\_services\_outlined*,  title: 'Doctor',  value: patient.doctorName,  ),  ),  Expanded(  child: \_buildInfoItem(  icon: Icons.*sick\_outlined*,  title: 'Symptoms',  value: patient.symptoms,  ),  ),  ],  ),  ],  ),  ),  ],  ),  );  }    Widget \_buildInfoItem({  required IconData icon,  required String title,  required dynamic value,  }) {  return Padding(  padding: const EdgeInsets.symmetric(vertical: 4),  child: Row(  crossAxisAlignment: CrossAxisAlignment.start,  children: [  Icon(  icon,  size: 16,  color: ThemeConstants.*textSecondaryColor*,  ),  const SizedBox(width: 8),  Expanded(  child: Column(  crossAxisAlignment: CrossAxisAlignment.start,  children: [  Text(  title.toString(),  style: const TextStyle(  fontSize: 12,  color: ThemeConstants.*textLightColor*,  ),  ),  const SizedBox(height: 2),  Text(  value.toString(),  style: const TextStyle(  color: ThemeConstants.*textPrimaryColor*,  fontSize: 14,  ),  maxLines: 2,  overflow: TextOverflow.ellipsis,  ),  ],  ),  ),  ],  ),  );  } }  **appointmentcontroller.dart**  import 'package:get/get.dart'; import 'package:lab\_ese/constants/doctors\_list.dart'; import 'package:lab\_ese/models/patient\_model.dart'; import 'package:lab\_ese/services/firebase\_service.dart';  class AppointmentController extends GetxController {  final FirebaseService \_firebaseService = FirebaseService();    final RxString name = ''.obs;  final RxString address = ''.obs;  final RxString age = ''.obs;  final RxString contactDetails = ''.obs;  final RxString symptoms = ''.obs;  final RxString selectedDoctor = doctors.first.name.obs;  final Rx<DateTime> appointmentDate = DateTime.now().obs;    final RxBool isLoading = false.obs;    final RxList<Patient> patients = <Patient>[].obs;    @override  void onInit() {  super.onInit();  loadPatients();  }    void setAppointmentDate(DateTime date) {  appointmentDate.value = date;  }    void setSelectedDoctor(String doctor) {  selectedDoctor.value = doctor;  }    Future<bool> saveAppointment() async {  isLoading.value = true;    try {  Patient patient = Patient(  name: name.value,  address: address.value,  age: int.*parse*(age.value),  contactDetails: int.*parse*(contactDetails.value),  symptoms: symptoms.value,  doctorName: selectedDoctor.value,  appointmentDate: appointmentDate.value,  );    await \_firebaseService.addPatient(patient);  resetForm();  await loadPatients();  return true;  } catch (e) {  print('Error saving appointment: $e');  return false;  } finally {  isLoading.value = false;  }  }    Future<void> loadPatients() async {  isLoading.value = true;    try {  final patientsList = await \_firebaseService.getPatients();  patients.value = patientsList;  } catch (e) {  print('Error loading patients: $e');  } finally {  isLoading.value = false;  }  }    void resetForm() {  name.value = '';  address.value = '';  age.value = '';  contactDetails.value = '';  symptoms.value = '';  selectedDoctor.value = doctors.first.name;  appointmentDate.value = DateTime.now();  } }  **Patientmodel.dart**  class Patient {  final String id;  final String name;  final String address;  final int age;  final int contactDetails;  final String symptoms;  final String doctorName;  final DateTime appointmentDate;   Patient({  this.id = '',  required this.name,  required this.address,  required this.age,  required this.contactDetails,  required this.symptoms,  required this.doctorName,  required this.appointmentDate,  });   Map<String, dynamic> toJson() {  return {  'name': name,  'address': address,  'age': age,  'contactDetails': contactDetails,  'symptoms': symptoms,  'doctorName': doctorName,  'appointmentDate': appointmentDate.toIso8601String(),  };  }   factory Patient.fromJson(Map<String, dynamic> json, String id) {  return Patient(  id: id,  name: json['name'] ?? '',  address: json['address'] ?? '',  age: json['age'] ?? 0,  contactDetails: json['contactDetails'] ?? '',  symptoms: json['symptoms'] ?? '',  doctorName: json['doctorName'] ?? '',  appointmentDate: json['appointmentDate'] != null  ? DateTime.*parse*(json['appointmentDate'])  : DateTime.now(),  );  } }  **doctorlist.dart**  class Doctor {  final String name;  final String specialization;   Doctor({  required this.name,  required this.specialization,  }); }  final List<Doctor> doctors = [  Doctor(name: 'Dr. Adam Ansari', specialization: 'Cardiologist'),  Doctor(name: 'Dr. Abhijeet Jadhav', specialization: 'Neurologist'),  Doctor(name: 'Dr. Vineet Shinde', specialization: 'Orthopedic'),  Doctor(name: 'Dr. Abhishek Jha', specialization: 'Pediatrician'),  Doctor(name: 'Dr. Ram Verma', specialization: 'Dermatologist'),  Doctor(name: 'Dr. Rohit Basak', specialization: 'Gynecologist'),  Doctor(name: 'Dr. Atharva Angre', specialization: 'General Physician'), ]; |
| **Screenshots** |  |
| **Question and Answers** | 1. Which Widgets you have used for the application?   TextFormField, showDatePicker, DropdownButtonFormField, ElevatedButton, Contaienr, Row, Column, etc.   1. Benefits of using flutter   It gives the power of Cross Platform development by using a single code base.   1. For the given theme which colour combination have you used and why?   I have used a shade of Blue, grey and white. As the topic is Doctor appointment booking, the UI must be user friendly, soothing, eye catchy and yet professional. |
| **Conclusion** | Implemented Firebase Connect as well as data handling, used 30-60-10 rule for UI designing, Proper Validation of the User Input fields. |