

GCARE CONNECT main.dart

```
1. import 'package:flutter/material.dart';
2. import 'package:intl/intl.dart';
3. import
  'package:shared_preferences/shared_preferences.dart';
4. import 'package:provider/provider.dart';
5. import 'package:google_fonts/google_fonts.dart';
6. import 'package:url_launcher/url_launcher.dart';
7. import 'package:fl_chart/fl_chart.dart';
8. import 'dart:convert';
9. import 'package:pdf/pdf.dart';
10. import 'package:pdf/widgets.dart' as pw;
11. import 'package:printing/printing.dart';
12.
13. // Text size provider for managing text scaling
14. class TextSizeProvider extends ChangeNotifier {
15.   double _textScaleFactor = 1.0;
16.
17.   double get textScaleFactor => _textScaleFactor;
18.
19.   void increaseTextSize() {
20.     if (_textScaleFactor < 1.3) {
21.       _textScaleFactor += 0.1;
22.       notifyListeners();
23.     }
24.   }
25.
26.   void resetTextSize() {
27.     _textScaleFactor = 1.0;
28.     notifyListeners();
29.   }
30.
31.   void decreaseTextSize() {
32.     if (_textScaleFactor > 0.7) {
33.       _textScaleFactor -= 0.1;
34.       notifyListeners();
35.     }
36.   }
37. }
38.
39. void main() async {
40.   WidgetsFlutterBinding.ensureInitialized();
41.   final authService = AuthService();
42.   await authService.loadAuthState();
43.   runApp(
44.     MultiProvider(
45.       providers: [
46.         ChangeNotifierProvider(create: (_) => authService),
47.         ChangeNotifierProvider(create: (_) =>
48.           TextSizeProvider()),
49.         ChangeNotifierProvider(create: (_) =>
50.           AppointmentService()),
51.         ChangeNotifierProvider(create: (_) =>
52.           FeedbackService()),
53.       ],
54.       child: GlobalCareApp(),
55.     );
56.   }
57.
58. class GlobalCareApp extends StatelessWidget {
59.   @override
60.   Widget build(BuildContext context) {
61.     return Consumer<TextSizeProvider>(
62.       builder: (context, textSizeProvider, child) {
63.         return MaterialApp(
64.           title: 'Global Care Medical Center',
65.           theme: ThemeData(
66.             primarySwatch: Colors.indigo,
67.             fontFamily: GoogleFonts.poppins().fontFamily,
68.             scaffoldBackgroundColor: Color(0xFF8FAFC),
69.             textTheme: TextTheme(
70.               headlineMedium: TextStyle(
71.                 fontSize: 28,
72.                 fontWeight: FontWeight.w600,
73.                 color: Color(0xFF1F2937),
74.               ),
75.               titleLarge: TextStyle(
76.                 fontSize: 18,
77.                 fontWeight: FontWeight.w500,
78.                 color: Color(0xFF1F2937),
79.               ),
80.               bodyMedium: TextStyle(
81.                 fontSize: 16,
82.                 fontWeight: FontWeight.w400,
83.                 color: Color(0xFF6B7280),
84.               ),
85.               labelLarge: TextStyle(
86.                 fontSize: 14,
87.                 fontWeight: FontWeight.w500,
88.                 color: Colors.white,
89.               ),
90.             appBarTheme: AppBarTheme(
91.               backgroundColor: Colors.white,
92.               elevation: 0,
93.               shadowColor: Colors.black12,
94.               iconTheme: IconThemeData(color:
95.                 Colors.indigo[900]),
96.             ),
97.             elevatedButtonTheme: ElevatedButtonThemeData(
98.               style: ElevatedButton.styleFrom(
99.                 backgroundColor: Colors.indigo,
100.                 foregroundColor: Colors.white,
101.                 elevation: 3,
102.                 shadowColor: Colors.indigo.withOpacity(0.3),
103.                 shape: RoundedRectangleBorder(
104.                   borderRadius: BorderRadius.circular(24),
105.                 ),
106.                 padding: EdgeInsets.symmetric(horizontal: 24,
107.                   vertical: 12),
108.               ),
109.               textStyle: TextStyle(
110.                 fontFamily: GoogleFonts.poppins().fontFamily,
111.                 fontSize: 16,
112.                 fontWeight: FontWeight.w500,
113.               ),
114.             ),
115.             textButtonTheme: TextButtonThemeData(
116.               style: TextButton.styleFrom(
117.                 foregroundColor: Colors.indigo,
118.                 backgroundColor: Colors.transparent,
119.                 shape: RoundedRectangleBorder(
120.                   borderRadius: BorderRadius.circular(24),
121.                 ),
122.                 padding: EdgeInsets.symmetric(horizontal: 16,
123.                   vertical: 8),
124.               ),
125.               minimumSize: Size(100, 36),
126.               textStyle: TextStyle(
127.                 fontFamily: GoogleFonts.poppins().fontFamily,
128.                 fontSize: 16,
129.                 fontWeight: FontWeight.w500,
130.               ),
131.             ),
132.           ),
133.         );
134.       },
135.     );
136.   }
137. }
```


GCARE CONNECT main.dart

```
250.         final tabController =
DefaultTabController.of(
251.             context,
252.         );
253.         if (tabController != null) {
254.             tabController.animateTo(3); //
Dashboard tab
255.             (context as
Element).markNeedsBuild();
256.         }
257.         } else {
258.             _showAuthDialog(context);
259.         }
260.     },
261.     tooltip: authService.isLoggedIn
262.       ? 'Profile'
263.       : 'Login/Sign Up',
264.   ),
265.   IconButton(
266.     icon: Icon(Icons.menu, color:
Colors.indigo[700]!),
267.     onPressed: () {
268.       Scaffold.of(context).openDrawer();
269.     },
270.   ),
271. ],
272. ),
273. ] else ...[
274.   Expanded(
275.     child: Center(
276.       child: Row(
277.         mainAxisAlignment: MainAxisAlignment.min,
278.         children: [
279.           _buildNavItem(
280.             context,
281.             'Home',
282.             0,
283.             DefaultTabController.of(context)?.index
== 0,
284.             isMobile,
285.           ),
286.           SizedBox(width: 24),
287.           _buildNavItem(
288.             context,
289.             'Appointments',
290.             1,
291.             DefaultTabController.of(context)?.index
== 1,
292.             isMobile,
293.           ),
294.           SizedBox(width: 24),
295.           _buildNavItem(
296.             context,
297.             'Feedback',
298.             2,
299.             DefaultTabController.of(context)?.index
== 2,
300.             isMobile,
301.           ),
302.           SizedBox(width: 24),
303.           _buildNavItem(
304.             context,
305.             'Dashboard',
306.             3,
307.             DefaultTabController.of(context)?.index
== 3,
308.             isMobile,
309.           ),
310.         ],
311.       ),
312.     ),
313.   ),
314.   Row(
315.     children: [
316.       if (!authService.isLoggedIn) ...[
317.         Container(
318.           decoration: BoxDecoration(
319.             border: Border.all(
320.               color: Colors.indigo[700]!,
321.               width: 1,
322.             ),
323.             borderRadius:
BorderRadius.circular(24),
324.           ),
325.           child: ElevatedButton(
326.             onPressed: () =>
_showAuthDialog(context),
327.             child: Text(
328.               'Login',
329.               style: TextStyle(
330.                 fontFamily:
GoogleFonts.poppins().fontFamily,
331.                 fontSize: 14, // Reduced font size
332.                 fontWeight: FontWeight.w500,
333.                 color: Colors.indigo[700],
334.               ),
335.             ),
336.             style: ElevatedButton.styleFrom(
337.               backgroundColor: Colors.white,
338.               shadowColor: Colors.transparent,
339.               padding: EdgeInsets.symmetric(
340.                 horizontal: isMobile
? 12
341.                 : 16, // Reduced padding
342.                 vertical: isMobile ? 6 : 8, // Reduced
padding
343.             ),
344.             shape: RoundedRectangleBorder(
345.               borderRadius:
BorderRadius.circular(24),
346.             ),
347.           ),
348.         ),
349.         ),
350.         ),
351.         ),
352.         ),
353.         ),
354.         ),
355.         ),
356.         ),
357.         ),
358.         ),
359.         ),
360.         ),
361.         ),
362.         ),
363.         ),
364.         ),
365.         ),
366.         ),
367.         ),
368.         ),
369.         ),
370.         ),
371.         ),
372.         ),
373.         ),
374.         ),
375.         ),
376.         ),
377.         ),
378.         ),
379.         ),
380.         ),
381.         ),
382.         ),
383.         ),
384.         ),
385.         ),
386.         ),
387.         ),
388.         ),
389.         ),
390.         ),
391.         ),
392.         ),
393.         ),
394.         ),
395.         ),
396.         ),
397.         ),
398.         ),
399.         ),
400.         ),
401.         ),
402.         ),
403.         ),
404.         ),
405.         ),
406.         ),
407.         ),
408.         ),
409.         ),
410.         ),
411.         ),
412.         ),
413.         ),
414.         ),
415.         ),
416.         ),
417.         ),
418.         ),
419.         ),
420.         ),
421.         ),
422.         ),
423.         ),
424.         ),
425.         ),
426.         ),
427.         ),
428.         ),
429.         ),
430.         ),
431.         ),
432.         ),
433.         ),
434.         ),
435.         ),
436.         ),
437.         ),
438.         ),
439.         ),
440.         ),
441.         ),
442.         ),
443.         ),
444.         ),
445.         ),
446.         ),
447.         ),
448.         ),
449.         ),
450.         ),
451.         ),
452.         ),
453.         ),
454.         ),
455.         ),
456.         ),
457.         ),
458.         ),
459.         ),
460.         ),
461.         ),
462.         ),
463.         ),
464.         ),
465.         ),
466.         ),
467.         ),
468.         ),
469.         ),
470.         ),
471.         ),
472.         ),
473.         ),
474.         ),
475.         ),
476.         ),
477.         ),
478.         ),
479.         ),
480.         ),
481.         ),
482.         ),
483.         ),
484.         ),
485.         ),
486.         ),
487.         ),
488.         ),
489.         ),
490.         ),
491.         ),
492.         ),
493.         ),
494.         ),
495.         ),
496.         ),
497.         ),
498.         ),
499.         ),
500.         ),
501.         ),
502.         ),
503.         ),
504.         ),
505.         ),
506.         ),
507.         ),
508.         ),
509.         ),
510.         ),
511.         ),
512.         ),
513.         ),
514.         ),
515.         ),
516.         ),
517.         ),
518.         ),
519.         ),
520.         ),
521.         ),
522.         ),
523.         ),
524.         ),
525.         ),
526.         ),
527.         ),
528.         ),
529.         ),
530.         ),
531.         ),
532.         ),
533.         ),
534.         ),
535.         ),
536.         ),
537.         ),
538.         ),
539.         ),
540.         ),
541.         ),
542.         ),
543.         ),
544.         ),
545.         ),
546.         ),
547.         ),
548.         ),
549.         ),
550.         ),
551.         ),
552.         ),
553.         ),
554.         ),
555.         ),
556.         ),
557.         ),
558.         ),
559.         ),
560.         ),
561.         ),
562.         ),
563.         ),
564.         ),
565.         ),
566.         ),
567.         ),
568.         ),
569.         ),
570.         ),
571.         ),
572.         ),
573.         ),
574.         ),
575.         ),
576.         ),
577.         ),
578.         ),
579.         ),
580.         ),
581.         ),
582.         ),
583.         ),
584.         ),
585.         ),
586.         ),
587.         ),
588.         ),
589.         ),
590.         ),
591.         ),
592.         ),
593.         ),
594.         ),
595.         ),
596.         ),
597.         ),
598.         ),
599.         ),
600.         ),
601.         ),
602.         ),
603.         ),
604.         ),
605.         ),
606.         ),
607.         ),
608.         ),
609.         ),
610.         ),
611.         ),
612.         ),
613.         ),
614.         ),
615.         ),
616.         ),
617.         ),
618.         ),
619.         ),
620.         ),
621.         ),
622.         ),
623.         ),
624.         ),
625.         ),
626.         ),
627.         ),
628.         ),
629.         ),
630.         ),
631.         ),
632.         ),
633.         ),
634.         ),
635.         ),
636.         ),
637.         ),
638.         ),
639.         ),
640.         ),
641.         ),
642.         ),
643.         ),
644.         ),
645.         ),
646.         ),
647.         ),
648.         ),
649.         ),
650.         ),
651.         ),
652.         ),
653.         ),
654.         ),
655.         ),
656.         ),
657.         ),
658.         ),
659.         ),
660.         ),
661.         ),
662.         ),
663.         ),
664.         ),
665.         ),
666.         ),
667.         ),
668.         ),
669.         ),
670.         ),
671.         ),
672.         ),
673.         ),
674.         ),
675.         ),
676.         ),
677.         ),
678.         ),
679.         ),
680.         ),
681.         ),
682.         ),
683.         ),
684.         ),
685.         ),
686.         ),
687.         ),
688.         ),
689.         ),
690.         ),
691.         ),
692.         ),
693.         ),
694.         ),
695.         ),
696.         ),
697.         ),
698.         ),
699.         ),
700.         ),
701.         ),
702.         ),
703.         ),
704.         ),
705.         ),
706.         ),
707.         ),
708.         ),
709.         ),
710.         ),
711.         ),
712.         ),
713.         ),
714.         ),
715.         ),
716.         ),
717.         ),
718.         ),
719.         ),
720.         ),
721.         ),
722.         ),
723.         ),
724.         ),
725.         ),
726.         ),
727.         ),
728.         ),
729.         ),
730.         ),
731.         ),
732.         ),
733.         ),
734.         ),
735.         ),
736.         ),
737.         ),
738.         ),
739.         ),
740.         ),
741.         ),
742.         ),
743.         ),
744.         ),
745.         ),
746.         ),
747.         ),
748.         ),
749.         ),
750.         ),
751.         ),
752.         ),
753.         ),
754.         ),
755.         ),
756.         ),
757.         ),
758.         ),
759.         ),
760.         ),
761.         ),
762.         ),
763.         ),
764.         ),
765.         ),
766.         ),
767.         ),
768.         ),
769.         ),
770.         ),
771.         ),
772.         ),
773.         ),
774.         ),
775.         ),
776.         ),
777.         ),
778.         ),
779.         ),
780.         ),
781.         ),
782.         ),
783.         ),
784.         ),
785.         ),
786.         ),
787.         ),
788.         ),
789.         ),
790.         ),
791.         ),
792.         ),
793.         ),
794.         ),
795.         ),
796.         ),
797.         ),
798.         ),
799.         ),
800.         ),
801.         ),
802.         ),
803.         ),
804.         ),
805.         ),
806.         ),
807.         ),
808.         ),
809.         ),
810.         ),
811.         ),
812.         ),
813.         ),
814.         ),
815.         ),
816.         ),
817.         ),
818.         ),
819.         ),
820.         ),
821.         ),
822.         ),
823.         ),
824.         ),
825.         ),
826.         ),
827.         ),
828.         ),
829.         ),
830.         ),
831.         ),
832.         ),
833.         ),
834.         ),
835.         ),
836.         ),
837.         ),
838.         ),
839.         ),
840.         ),
841.         ),
842.         ),
843.         ),
844.         ),
845.         ),
846.         ),
847.         ),
848.         ),
849.         ),
850.         ),
851.         ),
852.         ),
853.         ),
854.         ),
855.         ),
856.         ),
857.         ),
858.         ),
859.         ),
860.         ),
861.         ),
862.         ),
863.         ),
864.         ),
865.         ),
866.         ),
867.         ),
868.         ),
869.         ),
870.         ),
871.         ),
872.         ),
873.         ),
874.         ),
875.         ),
876.         ),
877.         ),
878.         ),
879.         ),
880.         ),
881.         ),
882.         ),
883.         ),
884.         ),
885.         ),
886.         ),
887.         ),
888.         ),
889.         ),
890.         ),
891.         ),
892.         ),
893.         ),
894.         ),
895.         ),
896.         ),
897.         ),
898.         ),
899.         ),
900.         ),
901.         ),
902.         ),
903.         ),
904.         ),
905.         ),
906.         ),
907.         ),
908.         ),
909.         ),
910.         ),
911.         ),
912.         ),
913.         ),
914.         ),
915.         ),
916.         ),
917.         ),
918.         ),
919.         ),
920.         ),
921.         ),
922.         ),
923.         ),
924.         ),
925.         ),
926.         ),
927.         ),
928.         ),
929.         ),
930.         ),
931.         ),
932.         ),
933.         ),
934.         ),
935.         ),
936.         ),
937.         ),
938.         ),
939.         ),
940.         ),
941.         ),
942.         ),
943.         ),
944.         ),
945.         ),
946.         ),
947.         ),
948.         ),
949.         ),
950.         ),
951.         ),
952.         ),
953.         ),
954.         ),
955.         ),
956.         ),
957.         ),
958.         ),
959.         ),
960.         ),
961.         ),
962.         ),
963.         ),
964.         ),
965.         ),
966.         ),
967.         ),
968.         ),
969.         ),
970.         ),
971.         ),
972.         ),
973.         ),
974.         ),
975.         ),
976.         ),
977.         ),
978.         ),
979.         ),
980.         ),
981.         ),
982.         ),
983.         ),
984.         ),
985.         ),
986.         ),
987.         ),
988.         ),
989.         ),
990.         ),
991.         ),
992.         ),
993.         ),
994.         ),
995.         ),
996.         ),
997.         ),
998.         ),
999.         ),
1000.         ),
```

GCARE CONNECT main.dart

```
366.         fontSize: 14, // Reduced font size
367.         fontWeight: FontWeight.w500,
368.         color: Colors.white,
369.       ),
370.     ),
371.     style: ElevatedButton.styleFrom(
372.       backgroundColor: Colors.indigo[700],
373.       shadowColor: Colors.transparent,
374.       padding: EdgeInsets.symmetric(
375.         horizontal: isMobile
376.           ? 12
377.           : 16, // Reduced padding
378.         vertical: isMobile ? 6 : 8, // Reduced
padding
379.       ),
380.       shape: RoundedRectangleBorder(
381.         borderRadius:
BorderRadius.circular(24),
382.       ),
383.     ),
384.   ),
385. ),
386. ] else ...[
387.   Row(
388.     children: [
389.       CircleAvatar(
390.         backgroundColor:
Colors.indigo.withOpacity(0.1),
391.         child: Icon(
392.           Icons.person,
393.           color: Colors.indigo[700]!,
394.           size: isMobile ? 20 : 24,
395.         ),
396.         radius: isMobile ? 14 : 16,
397.       ),
398.       SizedBox(width: 8),
399.       Text(
400.         '${authService.username}',
401.         style: GoogleFonts.poppins(
402.           color: Colors.indigo[700]!,
403.           fontSize: isMobile ? 14 : 16,
404.           fontWeight: FontWeight.w500,
405.         ),
406.       ),
407.     ],
408.   ),
409.   SizedBox(width: 8),
410.   IconButton(
411.     icon: Icon(
412.       Icons.logout,
413.       color: Colors.indigo[700]!,
414.     ),
415.     onPressed: () => authService.logout(),
416.   ),
417. ],
418. ],
419. ),
420. ],
421. ],
422. );
423. },
424. ),
425. ),
426. drawer: LayoutBuilder(
427.   builder: (context, constraints) {
428.     final isMobile = constraints.maxWidth < 600;
429.     if (!isMobile) return SizedBox.shrink();
430.     return Drawer(
431.       child: ListView(
432.         padding: EdgeInsets.zero,
433.         children: [
434.           DrawerHeader(
435.             decoration: BoxDecoration(color:
Colors.indigo[700]!),
436.             child: Row(
437.               children: [
438.                 Icon(Icons.favorite, color: Colors.white,
size: 32),
439.                 SizedBox(width: 8),
440.                 Text(
441.                   'Global Care',
442.                   style: GoogleFonts.poppins(
443.                     color: Colors.white,
444.                     fontSize: 24,
445.                     fontWeight: FontWeight.w700,
446.                   ),
447.                 ),
448.               ],
449.             ),
450.           ),
451.           ListTile(
452.             leading: Icon(Icons.home, color:
Colors.indigo[700]!),
453.             title: Text(
454.               'Home',
455.               style: GoogleFonts.poppins(
456.                 fontSize: 16,
457.                 color:
DefaultTabController.of(context)?.index == 0
458.                   ? Colors.indigo[700]
459.                   : Colors.black,
460.                 fontWeight:
DefaultTabController.of(context)?.index == 0
461.                   ? FontWeight.w600
462.                   : FontWeight.w400,
463.               ),
464.             ),
465.             onTap: () {
466.               DefaultTabController.of(context)?.animateTo(0);
467.               Navigator.pop(context);
468.               (context as Element).markNeedsBuild();
469.             },
470.           ),
471.           ListTile(
472.             leading: Icon(Icons.event, color:
Colors.indigo[700]!),
473.             title: Text(
474.               'Appointments',
475.               style: GoogleFonts.poppins(
476.                 fontSize: 16,
477.                 color:
DefaultTabController.of(context)?.index == 1
478.                   ? Colors.indigo[700]
479.                   : Colors.black,
480.                 fontWeight:
DefaultTabController.of(context)?.index == 1
481.                   ? FontWeight.w600
482.                   : FontWeight.w400,
483.               ),
484.             ),
485.             onTap: () {
```

GCARE CONNECT main.dart

```
486. DefaultTabController.of(context)?.animateTo(1);
487.     Navigator.pop(context);
488.     (context as Element).markNeedsBuild();
489.   },
490. ),
491. ListTile(
492.   leading: Icon(Icons.feedback, color:
Colors.indigo[700]!),
493.   title: Text(
494.     'Feedback',
495.     style: GoogleFonts.poppins(
496.       fontSize: 16,
497.       color:
DefaultTabController.of(context)?.index == 2
498.         ? Colors.indigo[700]
499.         : Colors.black,
500.       fontWeight:
DefaultTabController.of(context)?.index == 2
501.         ? FontWeight.w600
502.         : FontWeight.w400,
503.     ),
504.   ),
505.   onTap: () {
506.     DefaultTabController.of(context)?.animateTo(2);
507.     Navigator.pop(context);
508.     (context as Element).markNeedsBuild();
509.   },
510. ),
511. ListTile(
512.   leading: Icon(Icons.dashboard, color:
Colors.indigo[700]!),
513.   title: Text(
514.     'Dashboard',
515.     style: GoogleFonts.poppins(
516.       fontSize: 16,
517.       color:
DefaultTabController.of(context)?.index == 3
518.         ? Colors.indigo[700]
519.         : Colors.black,
520.       fontWeight:
DefaultTabController.of(context)?.index == 3
521.         ? FontWeight.w600
522.         : FontWeight.w400,
523.     ),
524.   ),
525.   onTap: () {
526.     DefaultTabController.of(context)?.animateTo(3);
527.     Navigator.pop(context);
528.     (context as Element).markNeedsBuild();
529.   },
530. ),
531. ],
532. ),
533. );
534. },
535. ),
536. body: TabBarView(
537.   children: [
538.     HomePage(),
539.     AppointmentPage(),
540.     FeedbackPage(),
541.     DashboardPage(),
542.   ],
543. ),
544. floatingActionButton: LayoutBuilder(
545.   builder: (context, constraints) {
546.     final isMobile = constraints.maxWidth < 600;
547.     return Container(
548.       padding: EdgeInsets.all(isMobile ? 6 : 8),
549.       decoration: BoxDecoration(
550.         color: Colors.white,
551.         borderRadius: BorderRadius.circular(24),
552.         boxShadow: [
553.           BoxShadow(
554.             color: Colors.black.withOpacity(0.1),
555.             blurRadius: 8,
556.             offset: Offset(0, 2),
557.           ),
558.         ],
559.       ),
560.       child: Row(
561.         mainAxisAlignment: MainAxisAlignment.min,
562.         children: [
563.           _buildFAB(
564.             heroTag: 'decreaseText',
565.             onPressed: textSizeProvider.decreaseTextSize,
566.             tooltip: 'Decrease text size',
567.             icon: Icons.text_decrease,
568.             isMobile: isMobile,
569.           ),
570.           _buildFAB(
571.             heroTag: 'resetText',
572.             onPressed: textSizeProvider.resetTextSize,
573.             tooltip: 'Reset text size',
574.             icon: Icons.text_fields,
575.             isMobile: isMobile,
576.           ),
577.           _buildFAB(
578.             heroTag: 'increaseText',
579.             onPressed: textSizeProvider.increaseTextSize,
580.             tooltip: 'Increase text size',
581.             icon: Icons.text_increase,
582.             isMobile: isMobile,
583.           ),
584.         ],
585.       ),
586.     );
587.   },
588. ),
589. floatingActionButtonLocation:
FloatingActionButtonLocation.endFloat,
590. ),
591. );
592. }
593.
594. Widget _buildFAB({
595.   required String heroTag,
596.   required VoidCallback onPressed,
597.   required String tooltip,
598.   required IconData icon,
599.   required bool isMobile,
600. }) {
601.   return IconButton(
602.     icon: Icon(icon, color: Colors.indigo[700]!, size: isMobile
? 18 : 20),
603.     onPressed: onPressed,
604.     tooltip: tooltip,
605.     padding: EdgeInsets.all(isMobile ? 6 : 8),
606.     constraints: BoxConstraints(),
```

GCARE CONNECT main.dart

```
607.   splashColor: Colors.blue[600]!,
608.   highlightColor: Colors.blue[400]!,
609. );
610. }
611. }
612.
613. class FeedbackPage extends StatefulWidget {
614.   @override
615.   State<FeedbackPage> createState() =>
616.     _FeedbackPageState();
617. }
618. class _FeedbackPageState extends State<FeedbackPage> {
619.   final _formKey = GlobalKey<FormState>();
620.   final feedbackController = TextEditingController();
621.   double satisfactionRating = 3.0;
622.   bool _isSubmitting = false;
623.
624.   void _submitFeedback() async {
625.     if (!_formKey.currentState!.validate()) {
626.       return;
627.     }
628.     setState(() => _isSubmitting = true);
629.     final feedback = {
630.       'rating': satisfactionRating,
631.       'comment': feedbackController.text,
632.       'date': DateFormat('MMM d,
633.         yyyy').format(DateTime.now()),
634.       'username':
635.         Provider.of<AuthService>(context, listen:
636.           false).username ?? 'Guest',
637.     };
638.     try {
639.       await Provider.of<FeedbackService>(
640.         context,
641.         listen: false,
642.       ).saveFeedback(feedback);
643.       ScaffoldMessenger.of(context).showSnackBar(
644.         SnackBar(
645.           content: Text('Feedback submitted successfully!'),
646.           backgroundColor: Colors.green,
647.           behavior: SnackBarBehavior.floating,
648.         ),
649.       );
650.       setState(() {
651.         feedbackController.clear();
652.         satisfactionRating = 3.0;
653.       });
654.     } catch (e) {
655.       ScaffoldMessenger.of(context).showSnackBar(
656.         SnackBar(
657.           content: Text('Failed to save feedback: $e'),
658.           backgroundColor: Colors.red,
659.           behavior: SnackBarBehavior.floating,
660.         ),
661.       );
662.     } finally {
663.       setState(() => _isSubmitting = false);
664.     }
665.   }
666.   @override
667.   Widget build(BuildContext context) {
668.     return Consumer<AuthService>(
669.       builder: (context, authService, child) {
670.         child: Column(
671.           children: [
672.             LayoutBuilder(
673.               builder: (context, constraints) {
674.                 final isMobile = constraints.maxWidth < 600;
675.                 return Container(
676.                   color: Color(0xFFF8FAFC),
677.                   padding: EdgeInsets.symmetric(
678.                     vertical: isMobile ? 32 : 48,
679.                     horizontal: constraints.maxWidth * 0.05,
680.                   ),
681.                   child: Column(
682.                     children: [
683.                       // Header
684.                       Column(
685.                         children: [
686.                           Text(
687.                             'Your Feedback Matters',
688.                             style: GoogleFonts.poppins(
689.                               fontSize: isMobile ? 26 : 30,
690.                               fontWeight: FontWeight.w700,
691.                               color: Color(0xFF1F2937),
692.                             ),
693.                           ),
694.                           Text(
695.                             'Help us improve by sharing your
696.                               experience.',
697.                             style: GoogleFonts.poppins(
698.                               fontSize: isMobile ? 15 : 16,
699.                               fontWeight: FontWeight.w400,
700.                               color: Color(0xFF6B7280),
701.                             ),
702.                           ),
703.                           Text(
704.                             // Feedback Form or Login Prompt
705.                             AnimatedOpacity(
706.                               opacity: 1.0,
707.                               duration: Duration(milliseconds: 500),
708.                               child: Card(
709.                                 elevation: 6,
710.                                 shape: RoundedRectangleBorder(
711.                                   borderRadius:
712.                                     BorderRadius.circular(16),
713.                                 ),
714.                                 child: Container(
715.                                   constraints: BoxConstraints(maxWidth:
716.                                     600),
717.                                   padding: EdgeInsets.all(isMobile ? 20 :
718.                                     24),
719.                                   decoration: BoxDecoration(
720.                                     color: Colors.white,
721.                                     borderRadius:
722.                                       BorderRadius.circular(16),
723.                                     boxShadow: [
724.                                       BoxShadow(
725.                                         color:
726.                                           Colors.black.withOpacity(0.05),
727.                                         blurRadius: 10,
728.                                         offset: Offset(0, 4),
729.                                       ),
730.                                     ],
731.                                   ),
732.                                 ),
733.                               ),
734.                             ),
735.                           ),
736.                         ],
737.                       ),
738.                     ],
739.                   ),
740.                 ),
741.               ),
742.             ),
743.           ],
744.         ),
745.       ),
746.     );
747.   }
748. }
```

GCARE CONNECT main.dart

```
730.         ),
731.         child: !AuthService.isLoggedInIn
732.           ? _buildLoginPrompt(isMobile)
733.           : _buildFeedbackForm(isMobile),
734.       ),
735.     ),
736.   ),
737.   // Added spacing before footer
738.   SizedBox(height: isMobile ? 32 : 48),
739. ],
740. ),
741. );
742. },
743. ),
744. // Footer now included in the scrollable content
745. FooterSection(),
746. ],
747. ),
748. );
749. },
750. );
751. }
752.
753. Widget _buildLoginPrompt(bool isMobile) {
754.   return Column(
755.     mainAxisAlignment: MainAxisAlignment.center,
756.     crossAxisAlignment: CrossAxisAlignment.center,
757.     children: [
758.       Icon(Icons.lock, size: isMobile ? 36 : 40, color:
759.         Colors.indigo[700]),
760.       SizedBox(height: 16),
761.       Text(
762.         'Sign in to share your feedback',
763.         style: GoogleFonts.poppins(
764.           fontSize: isMobile ? 18 : 20,
765.           fontWeight: FontWeight.w600,
766.           color: Color(0xFF1F2937)),
767.         ),
768.       ),
769.       SizedBox(height: 12),
770.       Text(
771.         'Please login or create an account to provide your
772.         valuable feedback.',
773.         style: GoogleFonts.poppins(
774.           fontSize: isMobile ? 14 : 15,
775.           color: Colors.grey[600],
776.         ),
777.       ),
778.       ),
779.       ElevatedButton(
780.         onPressed: () {
781.           showDialog(context: context, builder: (context) =>
782.             AuthDialog());
783.         },
784.         style: ElevatedButton.styleFrom(
785.           backgroundColor: Colors.indigo[700],
786.           foregroundColor: Colors.white,
787.           padding: EdgeInsets.symmetric(
788.             horizontal: isMobile ? 24 : 32,
789.             vertical: isMobile ? 12 : 14,
790.           ),
791.           shape: RoundedRectangleBorder(
792.             borderRadius: BorderRadius.circular(12),
793.           ),
794.           elevation: 2,
795.         ),
796.         child: Text(
797.           'Login / Sign Up',
798.           style: GoogleFonts.poppins(
799.             fontSize: isMobile ? 15 : 16,
800.             fontWeight: FontWeight.w600,
801.           ),
802.         ),
803.       ],
804.     );
805.   }
806.
807. Widget _buildFeedbackForm(bool isMobile) {
808.   return Form(
809.     key: _formKey,
810.     child: Column(
811.       crossAxisAlignment: CrossAxisAlignment.start,
812.       children: [
813.         Text(
814.           'Rate Your Experience',
815.           style: GoogleFonts.poppins(
816.             fontSize: isMobile ? 18 : 20,
817.             fontWeight: FontWeight.w600,
818.             color: Color(0xFF1F2937),
819.           ),
820.         ),
821.         SizedBox(height: 12),
822.         Row(
823.           mainAxisAlignment:
824.             MainAxisAlignment.spaceBetween,
825.           children: [
826.             Text(
827.               'Satisfaction Level:
828.               ${satisfactionRating.toStringAsFixed(1)}',
829.               style: GoogleFonts.poppins(
830.                 fontSize: isMobile ? 14 : 15,
831.                 color: Colors.grey[600],
832.               ),
833.             ),
834.             Row(
835.               children: List.generate(5, (index) {
836.                 return Icon(
837.                   index < satisfactionRating.floor()
838.                     ? Icons.star
839.                     : Icons.star_border,
840.                   color: Colors.amber,
841.                   size: isMobile ? 24 : 28,
842.                 );
843.               },
844.             ),
845.             ),
846.             ),
847.             ),
848.             ),
849.             ),
850.             ),
851.             ),
852.             ),
853.             ),
854.             ),
855.             ),
```

GCARE CONNECT main.dart

```
856.   SizedBox(height: 24),
857.   Text(
858.     'Your Feedback',
859.     style: GoogleFonts.poppins(
860.       fontSize: isMobile ? 18 : 20,
861.       fontWeight: FontWeight.w600,
862.       color: Color(0xFF1F2937),
863.     ),
864.   ),
865.   SizedBox(height: 12),
866.   AnimatedContainer(
867.     duration: Duration(milliseconds: 300),
868.     decoration: BoxDecoration(
869.       borderRadius: BorderRadius.circular(12),
870.       boxShadow: [
871.         BoxShadow(
872.           color: Colors.black.withOpacity(0.05),
873.           blurRadius: 6,
874.           offset: Offset(0, 2),
875.         ),
876.       ],
877.     ),
878.     child: TextFormField(
879.       controller: feedbackController,
880.       decoration: InputDecoration(
881.         labelText: 'Share your thoughts',
882.         hintText: 'Tell us about your experience or
suggestions...',
883.         border: OutlineInputBorder(
884.           borderRadius: BorderRadius.circular(12),
885.           borderSide: BorderSide.none,
886.         ),
887.         enabledBorder: OutlineInputBorder(
888.           borderRadius: BorderRadius.circular(12),
889.           borderSide: BorderSide.none,
890.         ),
891.         focusedBorder: OutlineInputBorder(
892.           borderRadius: BorderRadius.circular(12),
893.           borderSide: BorderSide(color:
Colors.indigo[700]!, width: 2),
894.         ),
895.         errorBorder: OutlineInputBorder(
896.           borderRadius: BorderRadius.circular(12),
897.           borderSide: BorderSide(color: Colors.red[400]!,
width: 1),
898.         ),
899.         focusedErrorBorder: OutlineInputBorder(
900.           borderRadius: BorderRadius.circular(12),
901.           borderSide: BorderSide(color: Colors.red[400]!,
width: 2),
902.         ),
903.         filled: true,
904.         fillColor: Colors.indigo.withOpacity(0.05),
905.         labelStyle: GoogleFonts.poppins(
906.           color: Colors.grey[600],
907.           fontSize: isMobile ? 14 : 15,
908.         ),
909.         hintStyle: GoogleFonts.poppins(
910.           color: Colors.grey[400],
911.           fontSize: isMobile ? 14 : 15,
912.         ),
913.       ),
914.       minLines: 4,
915.       maxLines: 6,
916.       validator: (value) {
917.         if (value == null || value.isEmpty) {
918.           return 'Please provide your feedback.';
919.         }
920.         return null;
921.       },
922.     ),
923.   ),
924.   SizedBox(height: 24),
925.   AnimatedContainer(
926.     duration: Duration(milliseconds: 300),
927.     width: double.infinity,
928.     child: ElevatedButton(
929.       onPressed: _isSubmitting ? null :
_submitFeedback,
930.       style: ElevatedButton.styleFrom(
931.         backgroundColor: Colors.indigo[700],
932.         foregroundColor: Colors.white,
933.         padding: EdgeInsets.symmetric(
934.           vertical: isMobile ? 14 : 16,
935.           horizontal: isMobile ? 24 : 32,
936.         ),
937.         shape: RoundedRectangleBorder(
938.           borderRadius: BorderRadius.circular(12),
939.         ),
940.         elevation: 2,
941.         shadowColor: Colors.indigo.withOpacity(0.3),
942.       ),
943.       child: _isSubmitting
? SizedBox(
944.         height: isMobile ? 20 : 24,
945.         width: isMobile ? 20 : 24,
946.         child: CircularProgressIndicator(
947.           strokeWidth: 2,
948.           valueColor:
AlwaysStoppedAnimation<Color>(Colors.white),
949.         ),
950.       )
951.     ) : Text(
952.       'Submit Feedback',
953.       style: GoogleFonts.poppins(
954.         fontSize: isMobile ? 15 : 16,
955.         fontWeight: FontWeight.w600,
956.       ),
957.     ),
958.   ),
959. ),
960. ),
961. ],
962. ),
963. );
964. }
965. }
966.
967. class HomePage extends StatelessWidget {
968.   @override
969.   Widget build(BuildContext context) {
970.     return SingleChildScrollView(
971.       child: Column(
972.         children: [
973.           HeroSection(),
974.           MissionVisionSection(),
975.           ServicesSection(),
976.           ContactSection(),
977.           FooterSection(),
978.         ],
979.       ),
980.     );
981.   }
```


GCARE CONNECT main.dart

```
982. }
983.
984. class AppointmentPage extends StatelessWidget {
985.   @override
986.   Widget build(BuildContext context) {
987.     return SingleChildScrollView(
988.       child: Column(children: [AppointmentSection(),
989.         FooterSection()]),
990.     );
991.   }
992.
993. class DashboardPage extends StatelessWidget {
994.   @override
995.   Widget build(BuildContext context) {
996.     return SingleChildScrollView(
997.       child: Column(children: [DashboardSection(),
998.         FooterSection()]),
999.     );
1000.   }
1001.
1002.   class AuthService extends ChangeNotifier {
1003.     bool _isLoggedIn = false;
1004.     String? _username;
1005.     String? _email;
1006.     static const String _usersKey = 'users';
1007.
1008.     bool get isLoggedIn => _isLoggedIn;
1009.     String? get username => _username;
1010.     String? get email => _email;
1011.
1012.     Future<void> loadAuthState() async {
1013.       final prefs = await
1014.         SharedPreferences.getInstance();
1015.       _isLoggedIn = prefs.getBool('isLoggedIn') ??
1016.         false;
1017.       _username = prefs.getString('username');
1018.       _email = prefs.getString('email');
1019.       notifyListeners();
1020.     }
1021.
1022.     Future<List<Map<String, dynamic>>>
1023.       _loadUsers() async {
1024.       final prefs = await
1025.         SharedPreferences.getInstance();
1026.       final usersJson = prefs.getString(_usersKey);
1027.       if (usersJson == null) return [];
1028.       return List<Map<String,
1029.         dynamic>>>.from(jsonDecode(usersJson));
1030.     }
1031.
1032.     Future<void> _saveUsers(List<Map<String,
1033.       dynamic>>> users) async {
1034.       final prefs = await
1035.         SharedPreferences.getInstance();
1036.       await prefs.setString(_usersKey,
1037.         jsonEncode(users));
1038.     }
1039.
1040.     Future<void> login(String email, String
1041.       password) async {
1042.       final prefs = await
1043.         SharedPreferences.getInstance();
1044.       final users = await _loadUsers();
1045.       final user = users.firstWhere(
1046.         (u) => u['email'] == email && u['password']
1047.         == password,
1048.         orElse: () => {},
1049.       );
1050.
1051.       if (user.isNotEmpty) {
1052.         _isLoggedIn = true;
1053.         _username = user['username'];
1054.         _email = email;
1055.         await prefs.setBool('isLoggedIn', true);
1056.         await prefs.setString('username',
1057.           user['username']);
1058.         await prefs.setString('email', email);
1059.         notifyListeners();
1060.       } else {
1061.         throw Exception('Invalid credentials');
1062.       }
1063.     }
1064.
1065.     Future<void> signup(String username, String
1066.       email, String password) async {
1067.       final prefs = await
1068.         SharedPreferences.getInstance();
1069.       final users = await _loadUsers();
1070.
1071.       if (users.any((u) => u['username'] ==
1072.         username)) {
1073.         throw Exception('Username already exists');
1074.       }
1075.
1076.       users.add({'username': username, 'email':
1077.         email, 'password': password});
1078.
1079.       await _saveUsers(users);
1080.       _isLoggedIn = true;
1081.       _username = username;
1082.       _email = email;
1083.       await prefs.setBool('isLoggedIn', true);
1084.       await prefs.setString('username', username);
1085.       await prefs.setString('email', email);
1086.       notifyListeners();
1087.     }
1088.
1089.     Future<void> logout() async {
1090.       final prefs = await
1091.         SharedPreferences.getInstance();
1092.       _isLoggedIn = false;
1093.       _username = null;
1094.       _email = null;
1095.       await prefs.setBool('isLoggedIn', false);
1096.       await prefs.remove('username');
1097.       await prefs.remove('email');
1098.       notifyListeners();
1099.     }
1100.   }
1101.
1102.   class AppointmentService extends ChangeNotifier
1103.   {
1104.     static const String _appointmentsKey =
1105.       'appointments';
1106.
1107.     Future<List<Map<String, dynamic>>>
1108.       loadAppointments() async {
1109.       final prefs = await
1110.         SharedPreferences.getInstance();
1111.     }
```

GCARE CONNECT main.dart

```
1090.         final appointmentsJson =
1091.           prefs.getString(_appointmentsKey);
1092.         if (appointmentsJson == null) return [];
1093.         return List<Map<String,
1094.           dynamic>>.from(jsonDecode(appointmentsJson));
1095.       }
1096.       Future<void> saveAppointment(Map<String,
1097.         dynamic> appointment) async {
1098.         final prefs = await
1099.           SharedPreferences.getInstance();
1100.         final appointments = await loadAppointments();
1101.         appointments.add(appointment);
1102.         await prefs.setString(_appointmentsKey,
1103.           jsonEncode(appointments));
1104.         notifyListeners();
1105.       }
1106.       Future<void> clearAppointments() async {
1107.         final prefs = await
1108.           SharedPreferences.getInstance();
1109.         await prefs.remove(_appointmentsKey);
1110.         notifyListeners();
1111.       }
1112.       class FeedbackService extends ChangeNotifier {
1113.         static const String _feedbackKey = 'feedback';
1114.         Future<List<Map<String, dynamic>>>
1115.           loadFeedback() async {
1116.             final prefs = await
1117.               SharedPreferences.getInstance();
1118.             final feedbackJson =
1119.               prefs.getString(_feedbackKey);
1120.             if (feedbackJson == null) return [];
1121.             return List<Map<String,
1122.               dynamic>>.from(jsonDecode(feedbackJson));
1123.           }
1124.           Future<void> saveFeedback(Map<String,
1125.             dynamic> feedback) async {
1126.             final prefs = await
1127.               SharedPreferences.getInstance();
1128.             final feedbackList = await loadFeedback();
1129.             feedbackList.add(feedback);
1130.             await prefs.setString(_feedbackKey,
1131.               jsonEncode(feedbackList));
1132.             notifyListeners();
1133.           }
1134.           Future<void> clearFeedback() async {
1135.             final prefs = await
1136.               SharedPreferences.getInstance();
1137.             await prefs.remove(_feedbackKey);
1138.             notifyListeners();
1139.           }
1140.           class AuthDialog extends StatefulWidget {
1141.             const AuthDialog({Key? key}) : super(key: key);
1142.             @override
1143.             State<AuthDialog> createState() =>
1144.               _AuthDialogState();
1145.           }
1146.           class _AuthDialogState extends
1147.             State<AuthDialog>
1148.             with SingleTickerProviderStateMixin {
1149.             late TabController _tabController;
1150.             final _loginFormKey = GlobalKey<FormState>();
1151.             final _signupFormKey =
1152.               GlobalKey<FormState>();
1153.             final _loginEmailController =
1154.               TextEditingController();
1155.             final _loginPasswordController =
1156.               TextEditingController();
1157.             final _signupUsernameController =
1158.               TextEditingController();
1159.             final _signupEmailController =
1160.               TextEditingController();
1161.             final _signupPasswordController =
1162.               TextEditingController();
1163.             final _confirmPasswordController =
1164.               TextEditingController();
1165.             @override
1166.             void initState() {
1167.               super.initState();
1168.               _tabController = TabController(length: 2,
1169.                 vsync: this);
1170.             }
1171.             @override
1172.             void dispose() {
1173.               _tabController.dispose();
1174.               _loginEmailController.dispose();
1175.               _loginPasswordController.dispose();
1176.               _signupUsernameController.dispose();
1177.               _signupEmailController.dispose();
1178.               _signupPasswordController.dispose();
1179.               _confirmPasswordController.dispose();
1180.               super.dispose();
1181.             }
1182.             void _submitLogin() async {
1183.               if (_loginFormKey.currentState!.validate()) {
1184.                 try {
1185.                   await Provider.of<AuthService>(
1186.                     context,
1187.                     listen: false,
1188.                     ).login(_loginEmailController.text,
1189.                       _loginPasswordController.text);
1190.                   Navigator.pop(context);
1191.                   ScaffoldMessenger.of(context).showSnackBar(
1192.                     const SnackBar(
1193.                       content: Text('Login successful!'),
1194.                       backgroundColor: Colors.green,
1195.                     ),
1196.                   );
1197.                 } catch (e) {
1198.                   ScaffoldMessenger.of(context).showSnackBar(
1199.                     SnackBar(
1200.                       content: Text('Login failed: $e'),
1201.                       backgroundColor: Colors.red,
1202.                     ),
1203.                   );
1204.                 }
1205.               }
1206.             }
1207.           }
1208.           }
1209.           }
1210.           }
1211.           }
1212.           }
1213.           }
1214.           }
1215.           }
1216.           }
1217.           }
1218.           }
1219.           }
1220.           }
1221.           }
1222.           }
1223.           }
1224.           }
1225.           }
1226.           }
1227.           }
1228.           }
1229.           }
1230.           }
1231.           }
1232.           }
1233.           }
1234.           }
1235.           }
1236.           }
1237.           }
1238.           }
1239.           }
1240.           }
1241.           }
1242.           }
1243.           }
1244.           }
1245.           }
1246.           }
1247.           }
1248.           }
1249.           }
1250.           }
1251.           }
1252.           }
1253.           }
1254.           }
1255.           }
1256.           }
1257.           }
1258.           }
1259.           }
1260.           }
1261.           }
1262.           }
1263.           }
1264.           }
1265.           }
1266.           }
1267.           }
1268.           }
1269.           }
1270.           }
1271.           }
1272.           }
1273.           }
1274.           }
1275.           }
1276.           }
1277.           }
1278.           }
1279.           }
1280.           }
1281.           }
1282.           }
1283.           }
1284.           }
1285.           }
1286.           }
1287.           }
1288.           }
1289.           }
1290.           }
1291.           }
1292.           }
1293.           }
1294.           }
1295.           }
1296.           }
1297.           }
1298.           }
1299.           }
1300.           }
```

GCARE CONNECT main.dart

```

1195.      }
1196.
1197.      void _submitSignup() async {
1198.        if (_signupFormKey.currentState!.validate()) {
1199.          if (_signupPasswordController.text !=
1200.            _confirmPasswordController.text) {
1201.            ScaffoldMessenger.of(context).showSnackBar(
1202.              const SnackBar(
1203.                content: Text('Passwords do not match'),
1204.                backgroundColor: Colors.red,
1205.              ),
1206.            );
1207.            return;
1208.          }
1209.          try {
1210.            await Provider.of<AuthService>(context,
1211.              listen: false).signup(
1212.                _signupUsernameController.text,
1213.                _signupEmailController.text,
1214.                _signupPasswordController.text,
1215.              );
1216.            Navigator.pop(context);
1217.            ScaffoldMessenger.of(context).showSnackBar(
1218.              const SnackBar(
1219.                content: Text('Sign up successful!'),
1220.                backgroundColor: Colors.green,
1221.              ),
1222.            );
1223.          } catch (e) {
1224.            ScaffoldMessenger.of(context).showSnackBar(
1225.              SnackBar(
1226.                content: Text('Sign up failed: $e'),
1227.                backgroundColor: Colors.red,
1228.              ),
1229.            );
1230.          }
1231.        }
1232.      @override
1233.      Widget build(BuildContext context) {
1234.        final isMobile =
1235.          MediaQuery.of(context).size.width < 600;
1236.        final theme = Theme.of(context);
1237.        return Dialog(
1238.          shape:
1239.            RoundedRectangleBorder(borderRadius:
1240.              BorderRadius.circular(16)),
1241.          insetPadding: EdgeInsets.symmetric(
1242.            horizontal: isMobile ? 16 : 24,
1243.            vertical: 24,
1244.          ),
1245.          child: ConstrainedBox(
1246.            constraints: BoxConstraints(
1247.              maxWidth: isMobile ? 400 : 500,
1248.              minWidth: isMobile ? 300 : 400,
1249.            ),
1250.            child: SingleChildScrollView(
1251.              child: Padding(
1252.                padding: EdgeInsets.all(isMobile ? 16 :
1253.                  24),
1254.                child: Column(
1255.                  mainAxisAlignment: MainAxisAlignment.min,
1256.                  children: [
1257.                    Row(
1258.                      mainAxisAlignment:
1259.                        MainAxisAlignment.center,
1260.                      children: [
1261.                        Icon(
1262.                          Icons.medical_services,
1263.                          color: theme.primaryColor,
1264.                          size: isMobile ? 24 : 32,
1265.                        ),
1266.                        SizedBox(width: 8),
1267.                        Text(
1268.                          'Global Care',
1269.                          style:
1270.                            theme.textTheme.titleLarge?.copyWith(
1271.                              color: theme.primaryColor,
1272.                              fontWeight: FontWeight.bold,
1273.                            ),
1274.                        ),
1275.                      ],
1276.                    ),
1277.                    SizedBox(height: 16),
1278.                    Container(
1279.                      decoration: BoxDecoration(
1280.                        color:
1281.                          theme.colorScheme.surfaceVariant.withOpacity(0.5),
1282.                        borderRadius:
1283.                          BorderRadius.circular(12),
1284.                      ),
1285.                      child: ListenableBuilder(
1286.                        listenable:
1287.                          _tabController.animation!,
1288.                        builder: (context, _) {
1289.                          final index =
1290.                            _tabController.animation!.value.round();
1291.                          return TabBar(
1292.                            controller: _tabController,
1293.                            labelColor: Colors.white,
1294.                            unselectedLabelColor:
1295.                              theme.primaryColor,
1296.                            indicator: BoxDecoration(
1297.                              color: theme.primaryColor,
1298.                              borderRadius: index == 0
1299.                                ? BorderRadius.only(
1300.                                  topLeft:
1301.                                    Radius.circular(12),
1302.                                  bottomLeft:
1303.                                    Radius.circular(12),
1304.                                )
1305.                                : BorderRadius.only(
1306.                                  topRight:
1307.                                    Radius.circular(12),
1308.                                  bottomRight:
1309.                                    Radius.circular(12),
1310.                                ),
1311.                            indicatorSize:
1312.                              TabBarIndicatorSize.tab,
1313.                            dividerColor: Colors.transparent,
1314.                            tabs: [
1315.                              Tab(
1316.                                child: Row(
1317.                                  mainAxisAlignment:
1318.                                    MainAxisAlignment.center,
1319.                                  children: [

```

GCARE CONNECT main.dart

1305.	Icon(Icons.login, size:	1360.)
	isMobile ? 18 : 20),	1361.	filled: true,
1306.	SizedBox(width: 4),	1362.	fillColor:
1307.	Text(Colors.indigo.withOpacity(0.05),
1308.	'Login',	1363.),
1309.	style:	1364.	validator: (value) {
	GoogleFonts.poppins(1365.	if (value == null
1310.	fontSize: isMobile ? 14 :	1366.	value.isEmpty
1311.	16,	1367.	!value.contains('@')) {
	fontWeight:	1368.	return 'Please enter a valid
	FontWeight.w500,		email';
1312.),	1369.	}
1313.),	1370.	return null;
1314.],	1371.	},
1315.),	1372.),
1316.),	1373.	SizedBox(height: 16),
1317.	Tab(1374.	TextFormField(
1318.	child: Row(1375.	controller:
1319.	mainAxisAlignment:		_loginPasswordController,
	MainAxisAlignment.center,	1376.	decoration: InputDecoration(
1320.	children: [1377.	labelText: 'Password',
1321.	Icon(1378.	prefixIcon: Icon(Icons.lock),
1322.	Icons.person_add,	1379.	border: OutlineInputBorder(
1323.	size: isMobile ? 18 : 20,	1380.	borderRadius:
1324.),		BorderRadius.circular(12),
1325.	SizedBox(width: 4),	1381.),
1326.	Text(1382.	filled: true,
1327.	'Sign Up',	1383.	fillColor:
1328.	style:		Colors.indigo.withOpacity(0.05),
	GoogleFonts.poppins(1384.),
1329.	fontSize: isMobile ? 14 :	1385.	obscureText: true,
1330.	16,	1386.	validator: (value) {
	fontWeight:	1387.	if (value == null
	FontWeight.w500,		value.isEmpty) {
1331.),	1388.	return 'Please enter
1332.),		password';
1333.],	1389.	}
1334.),	1390.	return null;
1335.),	1391.	},
1336.],	1392.),
1337.);	1393.	SizedBox(height: 16),
1338.	},	1394.	Align(
1339.),	1395.	alignment:
1340.),		Alignment.centerRight,
1341.	SizedBox(height: 16),	1396.	child: TextButton(
1342.	SizedBox(1397.	onPressed: () {
1343.	height: isMobile ? 400 : 450,	1398.	// Forgot password
1344.	child: TabBarView(1399.	},
1345.	controller: _tabController,	1400.	child: Text(
1346.	children: [1401.	'Forgot Password?',
1347.	// Login Tab	1402.	style:
1348.	Form(GoogleFonts.poppins(
1349.	key: _loginFormKey,	1403.	color:
1350.	child: Column(theme.primaryColor,
1351.	mainAxisSize: MainAxisSize.min,	1404.	fontSize: isMobile ? 12 :
1352.	children: [14,
1353.	TextFormField(1405.),
1354.	controller:	1406.),
	_loginEmailController,	1407.),
1355.	decoration: InputDecoration(1408.),
1356.	labelText: 'Email',	1409.	SizedBox(height: 16),
1357.	prefixIcon:	1410.	ElevatedButton(
	Icon(Icons.email),	1411.	onPressed: _submitLogin,
1358.	border: OutlineInputBorder(1412.	style:
1359.	borderRadius:		ElevatedButton.styleFrom(
	BorderRadius.circular(12),		

GCARE CONNECT main.dart

```

1413.      theme.primaryColor,
1414.      Colors.white,
1415.      Size(double.infinity, 50),
1416.      RoundedRectangleBorder(
1417.        BorderRadius.circular(12),
1418.      ),
1419.    ),
1420.    child: Text(
1421.      'Login',
1422.      style: GoogleFonts.poppins(
1423.        fontSize: isMobile ? 15 :
1424.        16,
1425.        FontWeight.w600,
1426.      ),
1427.    ),
1428.    SizedBox(height: 16),
1429.    Row(
1430.      mainAxisAlignment:
1431.      MainAxisAlignment.center,
1432.      children: [
1433.        Text(
1434.          "Don't have an account?",
1435.          style:
1436.          GoogleFonts.poppins(
1437.            fontSize: isMobile ? 12 :
1438.            14,
1439.            color: Colors.grey[600],
1440.          ),
1441.          TextButton(
1442.            onPressed: () =>
1443.            _tabController.animateTo(1),
1444.            child: Text(
1445.              'Sign Up',
1446.              style:
1447.              GoogleFonts.poppins(
1448.                color:
1449.                theme.primaryColor,
1450.                fontSize: isMobile ? 12 :
1451.                14,
1452.                fontWeight:
1453.                FontWeight.w600,
1454.              ),
1455.            ),
1456.          ],
1457.        ),
1458.        // Sign Up Tab
1459.        Form(
1460.          key: _signupFormKey,
1461.          child: Column(
1462.            mainAxisAlignment: MainAxisAlignment.min,
1463.            children: [
1464.              TextFormField(
1465.                decoration: InputDecoration(
1466.                  labelText: 'Username',
1467.                  prefixIcon:
1468.                  Icon(Icons.person),
1469.                  border: OutlineInputBorder(
1470.                    borderRadius:
1471.                    BorderRadius.circular(12),
1472.                  ),
1473.                  filled: true,
1474.                  fillColor:
1475.                  Colors.indigo.withOpacity(0.05),
1476.                ),
1477.                validator: (value) {
1478.                  if (value == null ||
1479.                      value.isEmpty) {
1480.                    return 'Please enter
1481.                      username';
1482.                  }
1483.                  return null;
1484.                },
1485.                ),
1486.              SizedBox(height: 16),
1487.              TextFormField(
1488.                controller:
1489.                _signupEmailController,
1490.                decoration: InputDecoration(
1491.                  labelText: 'Email',
1492.                  prefixIcon:
1493.                  Icon(Icons.email),
1494.                  border: OutlineInputBorder(
1495.                    borderRadius:
1496.                    BorderRadius.circular(12),
1497.                  ),
1498.                  filled: true,
1499.                  fillColor:
1500.                  Colors.indigo.withOpacity(0.05),
1501.                ),
1502.                validator: (value) {
1503.                  if (value == null ||
1504.                      value.isEmpty ||
1505.                      !value.contains('@')) {
1506.                    return 'Please enter valid
1507.                      email';
1508.                  }
1509.                  return null;
1510.                },
1511.                ),
1512.              SizedBox(height: 16),
1513.              TextFormField(
1514.                controller:
1515.                _signupPasswordController,
1516.                decoration: InputDecoration(
1517.                  labelText: 'Password',
1518.                  prefixIcon: Icon(Icons.lock),
1519.                  helperText: 'At least 6
1520.                    characters',
1521.                  border: OutlineInputBorder(
1522.                    borderRadius:
1523.                    BorderRadius.circular(12),
1524.                  ),
1525.                  filled: true,
1526.                  fillColor:
1527.                  Colors.indigo.withOpacity(0.05),
1528.                ),
1529.                obscureText: true,
1530.                validator: (value) {

```

GCARE CONNECT main.dart

1515.	if (value == null	1565.	children: [
1516.	value.isEmpty	1566.	Text(
1517.	value.length < 6) {	1567.	'Already have an
1518.	return 'Password must be		account?'
	at least 6 characters';	1568.	style:
1519.	}		GoogleFonts.poppins(
1520.	return null;	1569.	fontSize: isMobile ? 12 :
1521.	},		14,
1522.),	1570.	color: Colors.grey[600],
1523.	SizedBox(height: 16),	1571.),
1524.	TextFormField(1572.),
1525.	controller:	1573.	TextButton(
	_confirmPasswordController,	1574.	onPressed: () =>
1526.	decoration: InputDecoration(_tabController.animateTo(0),
1527.	labelText: 'Confirm	1575.	child: Text(
	Password',	1576.	'Login',
1528.	prefixIcon: Icon(Icons.lock),	1577.	style:
1529.	border: OutlineInputBorder(GoogleFonts.poppins(
1530.	borderRadius:	1578.	color:
	BorderRadius.circular(12),		theme.primaryColor,
1531.),	1579.	fontSize: isMobile ? 12 :
1532.	filled: true,		14,
1533.	fillColor:	1580.	fontWeight:
	Colors.indigo.withOpacity(0.05),		FontWeight.w600,
1534.),	1581.),
1535.	obscureText: true,	1582.),
1536.	validator: (value) {	1583.),
1537.	if (value == null	1584.],
	value.isEmpty) {	1585.),
1538.	return 'Please confirm	1586.],
	password';	1587.),
1539.	}	1588.),
1540.	return null;	1589.],
1541.	},	1590.),
1542.),	1591.),
1543.	SizedBox(height: 16),	1592.],
1544.	ElevatedButton(1593.),
1545.	onPressed: _submitSignup,	1594.),
1546.	style:	1595.),
	ElevatedButton.styleFrom(1596.),
1547.	theme.primaryColor,	1597.);
1548.	Colors.white,	1598.	}
	minimumSize:	1599.	}
1549.	Size(double.infinity, 50),	1600.	
1550.	shape:	1601.	class HeroSection extends StatelessWidget {
	RoundedRectangleBorder(1602.	const HeroSection({Key? key}) : super(key: key);
1551.	BorderRadius.circular(12),	1603.	
1552.),	1604.	@override
1553.),	1605.	Widget build(BuildContext context) {
1554.	child: Text(1606.	final authService =
1555.	'Create Account',		Provider.of<AuthService>(context);
1556.	style: GoogleFonts.poppins(1607.	
1557.	fontSize: isMobile ? 15 :	1608.	return LayoutBuilder(
	16,	1609.	builder: (context, constraints) {
1558.	fontWeight:	1610.	final isMobile = constraints.maxWidth < 600;
	FontWeight.w600,	1611.	final screenHeight =
1559.),		MediaQuery.of(context).size.height;
1560.),	1612.	
1561.),	1613.	return Stack(
1562.	SizedBox(height: 16),	1614.	children: [
1563.	Row(1615.	Container(
1564.	mainAxisAlignment:	1616.	height: screenHeight,
	MainAxisAlignment.center,	1617.	width: double.infinity,
		1618.	decoration: BoxDecoration(
		1619.	image: DecorationImage(
		1620.	image: NetworkImage(

GCARE CONNECT main.dart

```
1621. 'https://upload.wikimedia.org/wikipedia/commons/thumb/e/
    e1/2094Calamba_City_Canlubang_Roads_Landmarks_Baran
    gays_31.jpg/1200px-
    2094Calamba_City_Canlubang_Roads_Landmarks_Barangay
    s_31.jpg',
1622. ),
1623. fit: BoxFit.cover,
1624. ),
1625. ),
1626. ),
1627. Container(
1628.   height: screenHeight,
1629.   width: double.infinity,
1630.   decoration: BoxDecoration(
1631.     gradient: LinearGradient(
1632.       colors: [Color(0xAA1E40AF),
    Color(0x803B82F6)],
1633.     begin: Alignment.topLeft,
1634.     end: Alignment.bottomRight,
1635.   ),
1636. ),
1637. ),
1638. Container(
1639.   height: screenHeight,
1640.   padding: EdgeInsets.symmetric(
1641.     horizontal: constraints.maxWidth *
    0.05,
1642.     vertical: isMobile ? 16 : 24,
1643.   ),
1644.   alignment: Alignment.center,
1645.   child: Column(
1646.     mainAxisAlignment:
    MainAxisAlignment.center,
1647.     children: [
1648.       AnimatedOpacity(
1649.         opacity: 1.0,
1650.         duration: Duration(milliseconds:
    1000),
1651.         child: Text(
1652.           'Welcome to Global Care Medical
    Center',
1653.           textAlign: TextAlign.center,
1654.           style: GoogleFonts.poppins(
1655.             color: Colors.white,
1656.             fontSize: isMobile ? 24 : 36,
1657.             fontWeight: FontWeight.w700,
1658.             shadows: [
1659.               Shadow(
1660.                 color: Colors.black26,
1661.                 blurRadius: 4,
1662.                 offset: Offset(2, 2),
1663.               ),
1664.             ],
1665.           ),
1666.         ),
1667.       ),
1668.       SizedBox(height: isMobile ? 12 : 16),
1669.       AnimatedSlide(
1670.         offset: Offset(0, 0),
1671.         duration: Duration(milliseconds:
    1200),
1672.         curve: Curves.easeOut,
1673.         child: Text(
1674.           'Providing compassionate care and
    cutting-edge medical services to our community. At Global,
    We Care!',
1675.           textAlign: TextAlign.center,
1676.           style: GoogleFonts.poppins(
1677.             color:
    Colors.white.withOpacity(0.95),
1678.             fontSize: isMobile ? 14 : 18,
1679.             fontWeight: FontWeight.w400,
1680.           ),
1681.         ),
1682.       ),
1683.       SizedBox(height: isMobile ? 12 : 24),
1684.       Wrap(
1685.         alignment: WrapAlignment.center,
1686.         spacing: 12,
1687.         runSpacing: 8,
1688.         children: [
1689.           ElevatedButton(
1690.             style: ElevatedButton.styleFrom(
1691.               backgroundColor: Colors.white,
1692.               foregroundColor:
    Colors.indigo[700],
1693.             padding: EdgeInsets.symmetric(
1694.               horizontal: isMobile ? 20 : 28,
1695.               vertical: isMobile ? 12 : 16,
1696.             ),
1697.             shape:
    RoundedRectangleBorder(
1698.               borderRadius:
    BorderRadius.circular(24),
1699.             ),
1700.             elevation: 2,
1701.             shadowColor:
    Colors.black.withOpacity(0.2),
1702.           ),
1703.           onPressed: () {
1704.             if (authService.isLoggedIn) {
1705.               try {
1706.                 final tabController =
    DefaultTabController.of(
1707.                   context,
1708.                 );
1709.                 if (tabController != null) {
1710.                   tabController.animateTo(1);
1711.                   (context as
    Element).markNeedsBuild();
1712.                 } else {
1713.                   ScaffoldMessenger.of(context).showSnackBar(
1714.                     SnackBar(
1715.                       content: Text(
1716.                         'Navigation error.
    Please try again.',
1717.                       ),
1718.                     ),
1719.                   );
1720.                 }
1721.               } catch (e) {
1722.                 ScaffoldMessenger.of(context).showSnackBar(
1723.                   SnackBar(content:
    Text('Navigation error: $e')),
1724.                 );
1725.               }
1726.             } else {
```

GCARE CONNECT main.dart

```
1727.      showDialog(
1728.        context: context,
1729.        builder: (context) =>
1730.          AuthDialog(),
1731.      );
1732.    },
1733.    child: Text(
1734.      'Book Appointment',
1735.      style: GoogleFonts.poppins(
1736.        color: Colors.indigo[700],
1737.        fontSize: isMobile ? 14 : 16,
1738.        fontWeight: FontWeight.w600,
1739.      ),
1740.    ),
1741.  ),
1742.  OutlinedButton(
1743.    style: OutlinedButton.styleFrom(
1744.      backgroundColor:
1745.        Colors.transparent,
1746.      foregroundColor: Colors.white,
1747.      padding: EdgeInsets.symmetric(
1748.        horizontal: isMobile ? 20 : 28,
1749.        vertical: isMobile ? 12 : 16,
1750.      ),
1751.      side: BorderSide(color:
1752.        Colors.white, width: 1.5),
1753.      shape:
1754.        RoundedRectangleBorder(
1755.          borderRadius:
1756.            BorderRadius.circular(24),
1757.        ),
1758.      onPressed: () {
1759.        ScaffoldMessenger.of(context).showSnackBar(
1760.          SnackBar(content: Text('Learn
1761.            More clicked!')),
1762.        );
1763.      },
1764.      child: Text(
1765.        'Learn More',
1766.        style: GoogleFonts.poppins(
1767.          color: Colors.white,
1768.          fontSize: isMobile ? 14 : 16,
1769.          fontWeight: FontWeight.w500,
1770.        ),
1771.      ),
1772.    ),
1773.    SizedBox(height: isMobile ? 12 : 24),
1774.    HeroStats(isMobile: isMobile),
1775.  ],
1776. ),
1777. );
1778. },
1779. );
1780. }
1781. }
1782.
1783. class HeroStats extends StatelessWidget {
1784.   final bool isMobile;
1785.   HeroStats({required this.isMobile});
1786.
1787.   @override
1788.   Widget build(BuildContext context) {
1789.     return AnimatedOpacity(
1790.       opacity: 1.0,
1791.       duration: Duration(milliseconds: 1500),
1792.       child: Container(
1793.         margin: EdgeInsets.only(top: isMobile ? 12 :
1794.           24),
1795.         padding: EdgeInsets.symmetric(
1796.           horizontal: isMobile ? 12 : 24,
1797.           vertical: 12,
1798.         ),
1799.         decoration: BoxDecoration(
1800.           color: Colors.white.withOpacity(0.15),
1801.           borderRadius: BorderRadius.circular(16),
1802.           border: Border.all(color:
1803.             Colors.white.withOpacity(0.2)),
1804.           boxShadow: [BoxShadow(color:
1805.             Colors.black12, blurRadius: 6)],
1806.         ),
1807.         child: Wrap(
1808.           alignment: WrapAlignment.center,
1809.           spacing: isMobile ? 20 : 40,
1810.           runSpacing: 12,
1811.           children: [
1812.             StatItem(
1813.               icon: Icons.medical_services,
1814.               number: '210+',
1815.               label: 'Expert Doctors',
1816.               isMobile: isMobile,
1817.             ),
1818.             StatItem(
1819.               icon: Icons.local_hospital,
1820.               number: '150+',
1821.               label: 'Departments',
1822.               isMobile: isMobile,
1823.             ),
1824.             StatItem(
1825.               icon: Icons.people,
1826.               number: '10K+',
1827.               label: 'Happy Patients',
1828.               isMobile: isMobile,
1829.             ),
1830.           ],
1831.         ),
1832.       ),
1833.     );
1834.   }
1835. }
1836.
1837. class StatItem extends StatelessWidget {
1838.   final IconData icon;
1839.   final String number;
1840.   final String label;
1841.   final bool isMobile;
1842.   const StatItem({
1843.     required this.icon,
1844.     required this.number,
1845.     required this.label,
1846.     required this.isMobile,
1847.   });
1848.
1849.   @override
1850.   Widget build(BuildContext context) {
1851.     return Column(
1852.       mainAxisAlignment: MainAxisAlignment.center,
1853.       children: [
1854.         Icon(icon, size: isMobile ? 24 : 32, color: Colors.indigo[700]),
1855.         Text(number, style: TextStyle(fontWeight: FontWeight.bold, color: Colors.indigo[700])),
1856.         Text(label, style: TextStyle(color: Colors.indigo[700])),
1857.       ],
1858.     );
1859.   }
1860. }
```


GCARE CONNECT main.dart

```

1849.      children: [
1850.        Icon(icon, color: Colors.amber, size: isMobile
? 24 : 32),
1851.        SizedBox(height: 4),
1852.        Text(
1853.          number,
1854.          style: GoogleFonts.poppins(
1855.            color: Colors.white,
1856.            fontSize: isMobile ? 16 : 20,
1857.            fontWeight: FontWeight.w600,
1858.          ),
1859.        ),
1860.        Text(
1861.          label,
1862.          style: GoogleFonts.poppins(
1863.            color: Colors.white.withOpacity(0.9),
1864.            fontSize: isMobile ? 12 : 14,
1865.            fontWeight: FontWeight.w400,
1866.          ),
1867.        ),
1868.      ],
1869.    );
1870.  }
1871.}
1872.
1873. class MissionVisionSection extends StatelessWidget
{
1874.   @override
1875.   Widget build(BuildContext context) {
1876.     return LayoutBuilder(
1877.       builder: (context, constraints) {
1878.         final isMobile = constraints.maxWidth < 600;
1879.         return Container(
1880.           color: Color(0xFF8FAFC),
1881.           padding: EdgeInsets.symmetric(
1882.             vertical: isMobile ? 24 : 40,
1883.             horizontal: constraints.maxWidth * 0.05,
1884.           ),
1885.           child: Column(
1886.             children: [
1887.               SectionHeader(title: 'Our Mission &
Vision'),
1888.               SizedBox(height: isMobile ? 12 : 24),
1889.               isMobile
? Column(
1890.                 children: [
1891.                   MissionVisionCard(
1892.                     title: 'Our Mission',
1893.                     icon: Icons.track_changes,
1894.                     content:
1895.                       "To improve patient's lives by
providing safe, effective and suitable medical services
performed only by high capable staffs and medical
professionals while establishing an environment where
customers, employees and stakeholders are valued and
involved in continuously improving the quality of our services
towards a healthier community.",
1896.                     isMobile: isMobile,
1897.                   ),
1898.                   SizedBox(height: 16),
1899.                   MissionVisionCard(
1900.                     title: 'Our Vision',
1901.                     icon: Icons.remove_red_eye,
1902.                     content:
1903.                       "Global Care Medical Center
of Canlubang is the optimal healthcare provider in the
Community that delivers excellent medical service in the
most professional and compassionate way at the most
reasonable cost.",
1904.                     isMobile: isMobile,
1905.                   ),
1906.                 ],
1907.               ),
1908.             ],
1909.           );
1910.         }
1911.       );
1912.     }
1913.   }
1914. }
1915.
1916. "Global Care Medical Center
of Canlubang is the optimal healthcare provider in the
Community that delivers excellent medical service in the
most professional and compassionate way at the most
reasonable cost.",
1917.
1918. isMobile: isMobile,
1919. ),
1920. ),
1921. SizedBox(width: 24),
1922. Expanded(
1923.   child: MissionVisionCard(
1924.     title: 'Our Vision',
1925.     icon: Icons.remove_red_eye,
1926.     content:
1927.       "Global Care Medical Center
of Canlubang is the optimal healthcare provider in the
Community that delivers excellent medical service in the
most professional and compassionate way at the most
reasonable cost.",
1928.     isMobile: isMobile,
1929.   ),
1930. ),
1931. ],
1932. ),
1933. ],
1934. ),
1935. );
1936. },
1937. );
1938. }
1939. }
1940.
1941. class MissionVisionCard extends StatelessWidget {
1942.   final String title;
1943.   final IconData icon;
1944.   final String content;
1945.   final bool isMobile;
1946.
1947.   MissionVisionCard({
1948.     required this.title,
1949.     required this.icon,
1950.     required this.content,
1951.     required this.isMobile,
1952.   });
1953.
1954.   @override
1955.   Widget build(BuildContext context) {
1956.     return AnimatedOpacity(

```

GCARE CONNECT main.dart

```

1957.      opacity: 1.0,
1958.      duration: Duration(milliseconds: 800),
1959.      child: Card(
1960.        child: Padding(
1961.          padding: EdgeInsets.all(isMobile ? 16 : 24),
1962.          child: Column(
1963.            children: [
1964.              CircleAvatar(
1965.                backgroundColor:
1966.                  Colors.indigo.withOpacity(0.1),
1967.                child: Icon(icon, color:
1968.                  Colors.indigo[700]),
1969.                radius: isMobile ? 24 : 28,
1970.              ),
1971.              SizedBox(height: 12),
1972.              Text(
1973.                title,
1974.                style: GoogleFonts.poppins(
1975.                  fontSize: isMobile ? 16 : 18,
1976.                  fontWeight: FontWeight.w500,
1977.                ),
1978.              ),
1979.              SizedBox(height: 8),
1980.              Text(
1981.                content,
1982.                style: GoogleFonts.poppins(
1983.                  fontSize: isMobile ? 14 : 16,
1984.                  fontWeight: FontWeight.w400,
1985.                ),
1986.              ),
1987.              textAlign: TextAlign.center,
1988.            ],
1989.          ),
1990.        ),
1991.      );
1992.    }
1993.  }
1994.  class ServicesSection extends StatelessWidget {
1995.    final List<Map<String, dynamic>> services = [
1996.      {
1997.        'icon': Icons.local_hospital,
1998.        'title': 'Emergency Care',
1999.        'desc':
2000.          'Available 24/7 for all emergency medical
2001.          needs with state-of-the-art facilities.',
2002.      },
2003.      {
2004.        'icon': Icons.favorite,
2005.        'title': 'Primary Care',
2006.        'desc':
2007.          'Comprehensive primary healthcare services
2008.          for patients of all ages.',
2009.      },
2010.      {
2011.        'icon': Icons.healing,
2012.        'title': 'Specialized Treatment',
2013.        'desc':
2014.          'Expert specialists providing advanced care
2015.          across multiple medical disciplines.',
2016.      },
2017.      {
2018.        'icon': Icons.science,
2019.        'title': 'Laboratory Services',
2020.      },
2021.    ],
2022.  );
2023.  }
2024.  @override
2025.  Widget build(BuildContext context) {
2026.    return LayoutBuilder(
2027.      builder: (context, constraints) {
2028.        final isMobile = constraints.maxWidth < 600;
2029.        final crossAxisCount = constraints.maxWidth
2030.          < 600
2031.            ? 1
2032.            : constraints.maxWidth < 900
2033.              ? 2
2034.              : 3;
2035.        return Container(
2036.          color: Colors.white,
2037.          padding: EdgeInsets.symmetric(
2038.            vertical: isMobile ? 24 : 40,
2039.            horizontal: constraints.maxWidth * 0.05,
2040.          ),
2041.          child: Column(
2042.            children: [
2043.              SectionHeader(
2044.                title: 'Our Services',
2045.                subtitle:
2046.                  'Comprehensive medical services to
2047.                  meet all your healthcare needs.',
2048.              ),
2049.              SizedBox(height: isMobile ? 12 : 24),
2050.              GridView.count(
2051.                shrinkWrap: true,
2052.                physics: NeverScrollableScrollPhysics(),
2053.                crossAxisCount: crossAxisCount,
2054.                crossAxisSpacing: isMobile ? 16 : 24,
2055.                mainAxisSpacing: isMobile ? 16 : 24,
2056.                childAspectRatio: isMobile ? 1.3 : 1.1,
2057.                children: services
2058.                  .asMap()
2059.                  .entries
2060.                  .map(
2061.                    (entry) => AnimatedOpacity(
2062.                      opacity: 1.0,
2063.                      duration: Duration(milliseconds:
2064.                        500 + entry.key * 200),
2065.                      child: Card(
2066.                        child: Padding(
2067.                          padding:
2068.                            EdgeInsets.all(isMobile ? 16 : 20),
2069.                          child: Column(
2070.                            children: [

```

GCARE CONNECT main.dart

[illegible]

GCARE CONNECT main.dart

```

2188.      onTap: () => setState()
      => tabIndex = 1),
2189.      ),
2190.      ),
2191.      ],
2192.      ),
2193.      SizedBox(height: isMobile ? 16 :
2194.        24),
2195.      if (!AuthService.isLoggedIn &&
2196.        tabIndex == 0)
2197.      Column(
2198.        children: [
2199.          Icon(
2200.            Icons.lock,
2201.            size: isMobile ? 32 : 40,
2202.            color: Colors.indigo[700],
2203.          ),
2204.          SizedBox(height: 12),
2205.          Text(
2206.            'Please login or sign up to
2207.            book an appointment',
2208.            style:
2209.              GoogleFonts.poppins(
2210.                fontSize: isMobile ? 16 :
2211.                18,
2212.                fontWeight:
2213.                  FontWeight.w500,
2214.              ),
2215.              TextAlign.center,
2216.            ),
2217.            SizedBox(height: 16),
2218.            ElevatedButton(
2219.              onPressed: () {
2220.                showDialog(
2221.                  context: context,
2222.                  builder: (context) =>
2223.                    AuthDialog(),
2224.                );
2225.              },
2226.              child: Text('Login / Sign
2227.                Up'),
2228.            ),
2229.            ],
2230.          )
2231.        else
2232.          tabIndex == 0
2233.            ?
2234.              AppointmentForm(isMobile: isMobile)
2235.            :
2236.              AppointmentHistory(isMobile: isMobile),
2237.            ],
2238.          ),
2239.        ),
2240.      ],
2241.    );
2242.  },
2243.  ),
2244.  ],
2245.  ),
2246.  ),
2247.  ),
2248.  ),
2249.  ),
2250.  ),
2251.  ),
2252.  ),
2253.  ),
2254.  ),
2255.  ),
2256.  ),
2257.  ),
2258.  ),
2259.  ),
2260.  ),
2261.  ),
2262.  ),
2263.  ),
2264.  ),
2265.  ),
2266.  ),
2267.  ),
2268.  ),
2269.  ),
2270.  ),
2271.  ),
2272.  ),
2273.  ),
2274.  ),
2275.  ),
2276.  ),
2277.  ),
2278.  ),
2279.  ),
2280.  ),
2281.  ),
2282.  ),
2283.  ),
2284.  ),
2285.  ),
2286.  ),
2287.  ),
2288.  ),
2289.  ),
2290.  ),
2291.  ),
2292.  ),
2293.  ),
2294.  ),
2295.  ),
2296.  ),
2297.  ),
2298.  ),
2299.  ),
2300.  ),

```

GCARE CONNECT main.dart

```
2301.      'Neurology',
2302.      'Orthopedics',
2303.      'Pediatrics',
2304.      'Dermatology',
2305.    ];
2306.
2307.    final doctorsByDepartment = {
2308.      'Cardiology': [
2309.        'Dra. Joan Dympha P. Reano',
2310.        'Dr. Rocky Danillo S. Willis',
2311.        'Dr. James Patrick Diaz',
2312.      ],
2313.      'Neurology': [
2314.        'Dra. Cynthia B. Anacay',
2315.        'Dr. Mark Kryster I. Panaligan',
2316.        'Dr. Roman Santos',
2317.      ],
2318.      'Orthopedics': [
2319.        'Dra. Arabella P. Quing',
2320.        'Dra. Maria Thea M. Nido',
2321.        'Dr. Edgardo J. Cuadra',
2322.      ],
2323.      'Pediatrics': [
2324.        'Dr. Anner A. Marquez',
2325.        'Dr. Ching C. Casao',
2326.        'Dr. Jaypee V. Perez',
2327.      ],
2328.      'Dermatology': [
2329.        'Dra. Ashley R. Medina',
2330.        'Dra. Maria Reginas B. Evia',
2331.        'Dr. Gay C. San Antonio',
2332.      ],
2333.    };
2334.
2335.    final appointmentTypes = [
2336.      'New Patient',
2337.      'Follow-up',
2338.      'Consultation',
2339.      'Procedure',
2340.    ];
2341.
2342.    List<String> get doctors =>
2343.      department != null &&
2344.      doctorsByDepartment.containsKey(department)
2345.      ? doctorsByDepartment[department]!
2346.      : [];
2347.
2348.    List<String> get timeSlots {
2349.      final slots = <String>[];
2350.      for (int h = 9; h <= 17; h++) {
2351.        slots.add('${_formatHour(h)}:00
2352.          ${_ampm(h)}');
2353.        if (h < 17) slots.add('${_formatHour(h)}:30
2354.          ${_ampm(h)}');
2355.      }
2356.      return slots;
2357.    }
2358.
2359.    static String _formatHour(int h) => (h % 12 ==
2360.      0 ? 12 : h % 12).toString();
2361.    static String _ampm(int h) => h < 12 ? 'AM' :
2362.      'PM';
2363.
2364.    void _submit() {
2365.      String? errorMessage;
2366.      if (selectedDate == null) {
2367.        errorMessage = 'Please select a date.';
2368.      } else if (selectedTime == null) {
2369.        errorMessage = 'Please select a time slot.';
2370.      } else if (department == null) {
2371.        errorMessage = 'Please select a department.';
2372.      } else if (doctor == null) {
2373.        errorMessage = 'Please select a doctor.';
2374.      } else if (appointmentType == null) {
2375.        errorMessage = 'Please select an appointment
2376.          type.';
2377.      }
2378.
2379.      if (errorMessage != null) {
2380.        ScaffoldMessenger.of(
2381.          context,
2382.        ).showSnackBar(SnackBar(content:
2383.          Text(errorMessage)));
2384.        return;
2385.      }
2386.
2387.      final appointment = {
2388.        'avatar':
2389.          'https://api.dicebear.com/7.x/avataaars/svg?seed=${doctor!.
2390.            replaceAll(' ', '')}',
2391.        'doctor': doctor,
2392.        'department': department,
2393.        'type': appointmentType,
2394.        'date':
2395.          DateFormat('MMM d,
2396.            yyyy').format(selectedDate!) + ' at $selectedTime',
2397.        'status': 'Upcoming',
2398.        'reason': reasonController.text,
2399.      };
2400.
2401.      Provider.of<AppointmentService>(context,
2402.        listen: false)
2403.        .saveAppointment(appointment)
2404.        .then((_) {
2405.          if (mounted) {
2406.            ScaffoldMessenger.of(context).showSnackBar(
2407.              SnackBar(content: Text('Appointment
2408.                scheduled successfully!')),
2409.            );
2410.            Navigator.pushNamed(
2411.              context,
2412.              '/appointment-receipt',
2413.              arguments: {
2414.                'date': selectedDate,
2415.                'time': selectedTime,
2416.                'doctor': doctor,
2417.                'department': department,
2418.                'appointmentType': appointmentType,
2419.                'reason': reasonController.text,
2420.                'username':
2421.                  Provider.of<AuthService>(context,
2422.                    listen: false).username ??
2423.                    'Guest',
2424.                },
2425.            );
2426.            setState(() {
2427.              selectedDate = null;
2428.              selectedTime = null;
2429.              department = null;
2430.              doctor = null;
2431.            });
2432.          }
2433.        });
2434.    }
2435.  }
2436.}
```

GCARE CONNECT main.dart

```

2419.         appointmentType = null;
2420.         reasonController.clear();
2421.     });
2422.   }
2423. })
2424. .catchError((e) {
2425.   if (mounted) {
2426.     ScaffoldMessenger.of(context).showSnackBar(
2427.       SnackBar(content: Text('Failed to save
appointment: $e')),
2428.     );
2429.   }
2430. });
2431. }
2432.
2433. @override
2434. Widget build(BuildContext context) {
2435.   return Builder(
2436.     builder: (BuildContext) => widget.isMobile
2437.       ? Column(
2438.         crossAxisAlignment:
2439.           CrossAxisAlignment.start,
2440.         children: [
2441.           Text(
2442.             'Select Date & Time',
2443.             style: GoogleFonts.poppins(
2444.               fontSize: widget.isMobile ? 16 : 18,
2445.               fontWeight: FontWeight.w500,
2446.             ),
2447.           ),
2448.           SizedBox(height: 12),
2449.           CalendarDatePicker(
2450.             initialDate: DateTime.now(),
2451.             firstDate: DateTime.now(),
2452.             lastDate:
2453.               DateTime.now().add(Duration(days: 365)),
2454.             onChanged: (date) =>
2455.               setState(() {
2456.                 selectedDate = date;
2457.                 selectedTime = null;
2458.               }),
2459.           ),
2460.           SizedBox(height: 12),
2461.           Text(
2462.             'Available Time Slots',
2463.             style: GoogleFonts.poppins(
2464.               fontSize: widget.isMobile ? 16 : 18,
2465.               fontWeight: FontWeight.w500,
2466.             ),
2467.           ),
2468.           ),
2469.           SizedBox(height: 8),
2470.           Wrap(
2471.             spacing: 8,
2472.             runSpacing: 8,
2473.             children: timeSlots
2474.               .map(
2475.                 (slot) => ChoiceChip(
2476.                   label: Text(slot),
2477.                   selected: selectedTime == slot,
2478.                   onSelected: (_) =>
2479.                     setState(() => selectedTime
2480.                       = slot),
2481.                   selectedColor:
2482.                     Colors.indigo[700],

```

2477.	GoogleFonts.poppins(labelStyle:
2478.		color: selectedTime == slot
2479.		? Colors.white
2480.		: Colors.indigo[700],
2481.		fontSize: widget.isMobile ? 14
	: 16,	
2482.),
2483.		backgroundColor:
	Colors.indigo.withOpacity(0.05),	
2484.		side: BorderSide(color:
	Colors.indigo[700]!),	
2485.		padding: EdgeInsets.symmetric(
2486.		horizontal: widget.isMobile ?
	12 : 16,	
2487.		vertical: widget.isMobile ? 6 :
	8,	
2488.),
2489.		elevation: selectedTime == slot
	? 2 : 0,	
2490.),
2491.)
2492.		.toList(),
2493.),
2494.		SizeBox(height: 16),
2495.		Text(
2496.		'Appointment Details',
2497.		style: GoogleFonts.poppins(
2498.		fontSize: widget.isMobile ? 16 : 18,
2499.		fontWeight: FontWeight.w500,
2500.),
2501.),
2502.		SizeBox(height: 12),
2503.		DropDownButtonFormField<String>(
2504.		decoration: InputDecoration(
2505.		labelText: 'Department',
2506.		border: OutlineInputBorder(
2507.		borderRadius:
	BorderRadius.circular(12),	
2508.),
2509.		filled: true,
2510.		fillColor:
	Colors.indigo.withOpacity(0.05),	
2511.),
2512.		value: department,
2513.		items: [
2514.		DropDownMenuItem(
2515.		child: Text('Select Department'),
2516.		value: null,
2517.),
2518.		...departments.map(
2519.		(d) => DropDownMenuItem(child:
	Text(d, value: d),	
2520.),
2521.],
2522.		onChanged: (v) => setState(() {
2523.		department = v;
2524.		doctor = null;
2525.		}},
2526.),
2527.		SizeBox(height: 12),
2528.		DropDownButtonFormField<String>(
2529.		decoration: InputDecoration(
2530.		labelText: 'Doctor',
2531.		border: OutlineInputBorder(

GCARE CONNECT main.dart

```
2532.          borderRadius:
      BorderRadius.circular(12),
2533.        ),
2534.        filled: true,
2535.        fillColor:
      Colors.indigo.withOpacity(0.05),
2536.      ),
2537.      value: doctor,
2538.      items: [
2539.        DropdownMenuItem(child:
      Text('Select Doctor'), value: null),
2540.        ...doctors.map(
2541.          (d) => DropdownMenuItem(child:
      Text(d), value: d),
2542.        ),
2543.      ],
2544.      onChanged: (v) => setState(() =>
      doctor = v),
2545.    ),
2546.    SizedBox(height: 12),
2547.    DropdownButtonFormField<String>(
2548.      decoration: InputDecoration(
2549.        labelText: 'Appointment Type',
2550.        border: OutlineInputBorder(
2551.          borderRadius:
      BorderRadius.circular(12),
2552.        ),
2553.        filled: true,
2554.        fillColor:
      Colors.indigo.withOpacity(0.05),
2555.      ),
2556.      value: appointmentType,
2557.      items: [
2558.        DropdownMenuItem(child:
      Text('Select Type'), value: null),
2559.        ...appointmentTypes.map(
2560.          (t) => DropdownMenuItem(child:
      Text(t), value: t),
2561.        ),
2562.      ],
2563.      onChanged: (v) => setState(() =>
      appointmentType = v),
2564.    ),
2565.    SizedBox(height: 12),
2566.    TextFormField(
2567.      controller: reasonController,
2568.      decoration: InputDecoration(
2569.        labelText: 'Reason for Visit',
2570.        hintText:
      'Please describe your symptoms
      or reason for visit.',
2571.      ),
2572.      border: OutlineInputBorder(
2573.        borderRadius:
      BorderRadius.circular(12),
2574.      ),
2575.      filled: true,
2576.      fillColor:
      Colors.indigo.withOpacity(0.05),
2577.    ),
2578.    minLines: 3,
2579.    maxLines: 5,
2580.  ),
2581.  SizedBox(height: 16),
2582.  ElevatedButton(
2583.    onPressed: _submit,
2584.    child: Text('Schedule Appointment'),
2585.  ),
2586.  ],
2587. ),
2588. : Row(
2589.   crossAxisAlignment:
      CrossAxisAlignment.start,
2590.   children: [
2591.     Expanded(
2592.       child: Column(
2593.         crossAxisAlignment:
      CrossAxisAlignment.start,
2594.         children: [
2595.           Text(
2596.             'Select Date & Time',
2597.             style: GoogleFonts.poppins(
2598.               fontSize: 18,
2599.               fontWeight: FontWeight.w500,
2600.             ),
2601.           ),
2602.           SizedBox(height: 12),
2603.           CalendarDatePicker(
2604.             initialDate: DateTime.now(),
2605.             firstDate: DateTime.now(),
2606.             lastDate:
      DateTime.now().add(Duration(days: 365)),
2607.             onChanged: (date) =>
      setState() {
2608.               selectedDate = date;
2609.               selectedTime = null;
2610.             },
2611.           ),
2612.           SizedBox(height: 12),
2613.           Text(
2614.             'Available Time Slots',
2615.             style: GoogleFonts.poppins(
2616.               fontSize: 18,
2617.               fontWeight: FontWeight.w500,
2618.             ),
2619.           ),
2620.           SizedBox(height: 8),
2621.           Wrap(
2622.             spacing: 8,
2623.             runSpacing: 8,
2624.             children: timeSlots
      .map(
2625.               (slot) => ChoiceChip(
2626.                 label: Text(slot),
2627.                 selected: selectedTime ==
      slot,
2628.                 onSelected: (_) =>
      setState(() =>
      selectedTime = slot),
2629.               ),
2630.             ),
2631.           Colors.indigo[700],
2632.           GoogleFonts.poppins(
2633.             color: selectedTime ==
      slot
      ? Colors.white
      : Colors.indigo[700],
2634.           ),
2635.           ),
2636.           ),
2637.           Colors.indigo.withOpacity(
2638.             0.05,
2639.           ),
2640.         ],
2641.       ),
2642.     ),
2643.   ],
2644. ),
2645. ),
2646. ),
2647. ),
2648. ),
2649. ),
2650. ),
2651. ),
2652. ),
2653. ),
2654. ),
2655. ),
2656. ),
2657. ),
2658. ),
2659. ),
2660. ),
2661. ),
2662. ),
2663. ),
2664. ),
2665. ),
2666. ),
2667. ),
2668. ),
2669. ),
2670. ),
2671. ),
2672. ),
2673. ),
2674. ),
2675. ),
2676. ),
2677. ),
2678. ),
2679. ),
2680. ),
2681. ),
2682. ),
2683. ),
2684. ),
2685. ),
2686. ),
2687. ),
2688. ),
2689. ),
2690. ),
2691. ),
2692. ),
2693. ),
2694. ),
2695. ),
2696. ),
2697. ),
2698. ),
2699. ),
2700. ),
2701. ),
2702. ),
2703. ),
2704. ),
2705. ),
2706. ),
2707. ),
2708. ),
2709. ),
2710. ),
2711. ),
2712. ),
2713. ),
2714. ),
2715. ),
2716. ),
2717. ),
2718. ),
2719. ),
2720. ),
2721. ),
2722. ),
2723. ),
2724. ),
2725. ),
2726. ),
2727. ),
2728. ),
2729. ),
2730. ),
2731. ),
2732. ),
2733. ),
2734. ),
2735. ),
2736. ),
2737. ),
2738. ),
2739. ),
2740. ),
2741. ),
2742. ),
2743. ),
2744. ),
2745. ),
2746. ),
2747. ),
2748. ),
2749. ),
2750. ),
2751. ),
2752. ),
2753. ),
2754. ),
2755. ),
2756. ),
2757. ),
2758. ),
2759. ),
2760. ),
2761. ),
2762. ),
2763. ),
2764. ),
2765. ),
2766. ),
2767. ),
2768. ),
2769. ),
2770. ),
2771. ),
2772. ),
2773. ),
2774. ),
2775. ),
2776. ),
2777. ),
2778. ),
2779. ),
2780. ),
2781. ),
2782. ),
2783. ),
2784. ),
2785. ),
2786. ),
2787. ),
2788. ),
2789. ),
2790. ),
2791. ),
2792. ),
2793. ),
2794. ),
2795. ),
2796. ),
2797. ),
2798. ),
2799. ),
2800. ),
2801. ),
2802. ),
2803. ),
2804. ),
2805. ),
2806. ),
2807. ),
2808. ),
2809. ),
2810. ),
2811. ),
2812. ),
2813. ),
2814. ),
2815. ),
2816. ),
2817. ),
2818. ),
2819. ),
2820. ),
2821. ),
2822. ),
2823. ),
2824. ),
2825. ),
2826. ),
2827. ),
2828. ),
2829. ),
2830. ),
2831. ),
2832. ),
2833. ),
2834. ),
2835. ),
2836. ),
2837. ),
2838. ),
2839. ),
2840. ),
2841. ),
2842. ),
2843. ),
2844. ),
2845. ),
2846. ),
2847. ),
2848. ),
2849. ),
2850. ),
2851. ),
2852. ),
2853. ),
2854. ),
2855. ),
2856. ),
2857. ),
2858. ),
2859. ),
2860. ),
2861. ),
2862. ),
2863. ),
2864. ),
2865. ),
2866. ),
2867. ),
2868. ),
2869. ),
2870. ),
2871. ),
2872. ),
2873. ),
2874. ),
2875. ),
2876. ),
2877. ),
2878. ),
2879. ),
2880. ),
2881. ),
2882. ),
2883. ),
2884. ),
2885. ),
2886. ),
2887. ),
2888. ),
2889. ),
2890. ),
2891. ),
2892. ),
2893. ),
2894. ),
2895. ),
2896. ),
2897. ),
2898. ),
2899. ),
2900. ),
2901. ),
2902. ),
2903. ),
2904. ),
2905. ),
2906. ),
2907. ),
2908. ),
2909. ),
2910. ),
2911. ),
2912. ),
2913. ),
2914. ),
2915. ),
2916. ),
2917. ),
2918. ),
2919. ),
2920. ),
2921. ),
2922. ),
2923. ),
2924. ),
2925. ),
2926. ),
2927. ),
2928. ),
2929. ),
2930. ),
2931. ),
2932. ),
2933. ),
2934. ),
2935. ),
2936. ),
2937. ),
2938. ),
2939. ),
2940. ),
2941. ),
2942. ),
2943. ),
2944. ),
2945. ),
2946. ),
2947. ),
2948. ),
2949. ),
2950. ),
2951. ),
2952. ),
2953. ),
2954. ),
2955. ),
2956. ),
2957. ),
2958. ),
2959. ),
2960. ),
2961. ),
2962. ),
2963. ),
2964. ),
2965. ),
2966. ),
2967. ),
2968. ),
2969. ),
2970. ),
2971. ),
2972. ),
2973. ),
2974. ),
2975. ),
2976. ),
2977. ),
2978. ),
2979. ),
2980. ),
2981. ),
2982. ),
2983. ),
2984. ),
2985. ),
2986. ),
2987. ),
2988. ),
2989. ),
2990. ),
2991. ),
2992. ),
2993. ),
2994. ),
2995. ),
2996. ),
2997. ),
2998. ),
2999. ),
3000. ),
```

GCARE CONNECT main.dart

```
2640.      side: BorderSide(color:
Colors.indigo[700]!),
2641.      padding:
EdgeInsets.symmetric(
2642.        horizontal: 16,
2643.        vertical: 8,
2644.      ),
2645.      elevation: selectedTime ==
slot ? 2 : 0,
2646.    ),
2647.  )
2648.    .toList(),
2649.  ),
2650. ],
2651. ),
2652. ),
2653.   SizedBox(width: 24),
2654.   Expanded(
2655.     child: Column(
2656.       crossAxisAlignment:
CrossAxisAlignment.start,
2657.       children: [
2658.         Text(
2659.           'Appointment Details',
2660.           style: GoogleFonts.poppins(
2661.             fontSize: 18,
2662.             fontWeight: FontWeight.w500,
2663.           ),
2664.         ),
2665.         SizedBox(height: 12),
2666.         DropdownButtonFormField<String>(
2667.           decoration: InputDecoration(
2668.             labelText: 'Department',
2669.             border: OutlineInputBorder(
2670.               borderRadius:
BorderRadius.circular(12),
2671.             ),
2672.             filled: true,
2673.             fillColor:
Colors.indigo.withOpacity(0.05),
2674.           ),
2675.           value: department,
2676.           items: [
2677.             DropdownMenuItem(
2678.               child: Text('Select
Department'),
2679.               value: null,
2680.             ),
2681.             ...departments.map(
2682.               (d) =>
DropdownMenuItem(child: Text(d), value: d),
2683.             ),
2684.           ],
2685.           onChanged: (v) => setState() {
2686.             department = v;
2687.             doctor = null;
2688.           },
2689.         ),
2690.         SizedBox(height: 12),
2691.         DropdownButtonFormField<String>(
2692.           decoration: InputDecoration(
2693.             labelText: 'Doctor',
2694.             border: OutlineInputBorder(
2695.               borderRadius:
BorderRadius.circular(12),
2696.             ),
2697.             filled: true,
2698.             fillColor:
Colors.indigo.withOpacity(0.05),
2699.           ),
2700.           value: doctor,
2701.           items: [
2702.             DropdownMenuItem(
2703.               child: Text('Select Doctor'),
2704.               value: null,
2705.             ),
2706.             ...doctors.map(
2707.               (d) =>
DropdownMenuItem(child: Text(d), value: d),
2708.             ),
2709.           ],
2710.           onChanged: (v) => setState()
=> doctor = v),
2711.         ),
2712.         SizedBox(height: 12),
2713.         DropdownButtonFormField<String>(
2714.           decoration: InputDecoration(
2715.             labelText: 'Appointment Type',
2716.             border: OutlineInputBorder(
2717.               borderRadius:
BorderRadius.circular(12),
2718.             ),
2719.             filled: true,
2720.             fillColor:
Colors.indigo.withOpacity(0.05),
2721.           ),
2722.           value: appointmentType,
2723.           items: [
2724.             DropdownMenuItem(
2725.               child: Text('Select Type'),
2726.               value: null,
2727.             ),
2728.             ...appointmentTypes.map(
2729.               (t) =>
DropdownMenuItem(child: Text(t), value: t),
2730.             ),
2731.           ],
2732.           onChanged: (v) => setState()
=> appointmentType = v),
2733.         ),
2734.         SizedBox(height: 12),
2735.         TextFormField(
2736.           controller: reasonController,
2737.           decoration: InputDecoration(
2738.             labelText: 'Reason for Visit',
2739.             hintText:
'Please describe your
symptoms or reason for visit.',
2740.             border: OutlineInputBorder(
2741.               borderRadius:
BorderRadius.circular(12),
2742.             ),
2743.           ),
2744.           filled: true,
2745.           fillColor:
Colors.indigo.withOpacity(0.05),
2746.         ),
2747.         minLines: 3,
2748.         maxLines: 5,
```


GCARE CONNECT main.dart

```
2749.           ),
2750.           SizedBox(height: 16),
2751.           ElevatedButton(
2752.             onPressed: _submit,
2753.             child: Text('Schedule
Appointment'),
2754.           ),
2755.         ],
2756.       ),
2757.     ),
2758.   ],
2759. ),
2760. );
2761. }
2762. }
2763.
2764. class AppointmentReceiptPage extends
StatelessWidget {
2765.   Future<void> _downloadReceipt(
2766.     BuildContext context, {
2767.     required String id,
2768.     required String? username,
2769.     required DateTime? date,
2770.     required String? time,
2771.     required String? doctor,
2772.     required String? department,
2773.     required String? appointmentType,
2774.     required String? reason,
2775.   }) async {
2776.     final pdf = pw.Document();
2777.     final font = await
PdfGoogleFonts.poppinsRegular();
2778.
2779.     pdf.addPage(
2780.       pw.Page(
2781.         pageFormat: PdfPageFormat.a4,
2782.         build: (pw.Context context) {
2783.           return pw.Padding(
2784.             padding: const pw.EdgeInsets.all(24),
2785.             child: pw.Column(
2786.               crossAxisAlignment:
pw.CrossAxisAlignment.start,
2787.               children: [
2788.                 pw.Row(
2789.                   mainAxisAlignment:
pw.MainAxisAlignment.center,
2790.                   children: [
2791.                     pw.Text(
2792.                       'Global Care Medical Center',
2793.                       style: pw.TextStyle(
2794.                         font: font,
2795.                         fontSize: 24,
2796.                         fontWeight: pw.FontWeight.bold,
2797.                         color: PdfColors.indigo900,
2798.                       ),
2799.                     ),
2800.                   ],
2801.                 ),
2802.                 pw.SizedBox(height: 16),
2803.                 pw.Text(
2804.                   'Appointment Receipt',
2805.                   style: pw.TextStyle(
2806.                     font: font,
2807.                     fontSize: 20,
2808.                     fontWeight: pw.FontWeight.bold,
2809.                     color: PdfColors.indigo700,
2810.                   ),
2811.                 ),
2812.                 pw.SizedBox(height: 8),
2813.                 pw.Text(
2814.                   'Your appointment has been
successfully scheduled.',
2815.                   style: pw.TextStyle(
2816.                     font: font,
2817.                     fontSize: 14,
2818.                     color: PdfColors.grey600,
2819.                   ),
2820.                 ),
2821.                 pw.SizedBox(height: 24),
2822.                 pw.Container(
2823.                   padding: const pw.EdgeInsets.all(12),
2824.                   decoration: pw.BoxDecoration(
2825.                     border: pw.Border.all(color:
PdfColors.indigo200),
2826.                     borderRadius:
pw.BorderRadius.circular(12),
2827.                   ),
2828.                   child: pw.Column(
2829.                     crossAxisAlignment:
pw.CrossAxisAlignment.start,
2830.                     children: [
2831.                       pw.Text(
2832.                         'Appointment Details',
2833.                         style: pw.TextStyle(
2834.                           font: font,
2835.                           fontSize: 16,
2836.                           fontWeight:
pw.FontWeight.bold,
2837.                         ),
2838.                       ),
2839.                       pw.SizedBox(height: 8),
2840.                       _pdfSummaryRow('Appointment
ID:', '#$id', font),
2841.                       _pdfSummaryRow('Patient:',
username ?? 'Guest', font),
2842.                       _pdfSummaryRow(
'Date:',
date != null
? DateFormat('MMMM d,
yyyy').format(date)
: 'N/A',
font),
2843.                       _pdfSummaryRow('Time:', time ??
'N/A', font),
2844.                       _pdfSummaryRow('Doctor:', doctor
?? 'N/A', font),
2845.                       _pdfSummaryRow('Department:',
department ?? 'N/A', font),
2846.                       _pdfSummaryRow('Type:',
appointmentType ?? 'N/A', font),
2847.                       _pdfSummaryRow(
'Reason:',
reason?.isEmpty ?? true ? 'Not
specified' : reason!,
font),
2848.                     ],
2849.                   ),
2850.                 pw.SizedBox(height: 16),
2851.                 pw.Container(
```

GCARE CONNECT main.dart

```
2863.          padding: const
          pw.EdgeInsets.only(left: 12),
2864.          decoration: const pw.BoxDecoration(
2865.            border: pw.Border(
2866.              left: pw.BorderSide(color:
                PdfColors.indigo700, width: 3),
2867.            ),
2868.          ),
2869.          child: pw.Column(
2870.            crossAxisAlignment:
                pw.CrossAxisAlignment.start,
2871.            children: [
2872.              pw.Text(
2873.                'Confirmation',
2874.                style: pw.TextStyle(
2875.                  font: font,
2876.                  fontSize: 16,
2877.                  fontWeight:
                    pw.FontWeight.bold,
2878.                ),
2879.              ),
2880.              pw.SizedBox(height: 8),
2881.              pw.Text(
2882.                'A confirmation email will be sent
                to your registered email address.',
2883.                style: pw.TextStyle(font: font,
                fontSize: 12),
2884.              ),
2885.              pw.SizedBox(height: 8),
2886.              pw.Text(
2887.                'Please arrive 15 minutes early
                and bring a valid ID.',
2888.                style: pw.TextStyle(font: font,
                fontSize: 12),
2889.              ),
2890.              pw.SizedBox(height: 8),
2891.              pw.Text(
2892.                'Location: J. Yulo Avenue, Brgy.
                Canlubang',
2893.                style: pw.TextStyle(font: font,
                fontSize: 12),
2894.              ),
2895.            ],
2896.          ),
2897.        ),
2898.      ],
2899.    ),
2900.  );
2901. },
2902. ),
2903. );
2904.
2905. await Printing.sharePdf(
2906.   bytes: await pdf.save(),
2907.   filename: 'appointment_receipt_${id}.pdf',
2908. );
2909. }
2910.
2911. pw.Widget _pdfSummaryRow(String label, String
  value, pw.Font font) {
2912.   return pw.Padding(
2913.     padding: const
2914.       pw.EdgeInsets.symmetric(vertical: 4),
2915.     child: pw.Row(
2916.       crossAxisAlignment:
2917.         pw.CrossAxisAlignment.start,
2918.       children: [
2919.         pw.SizedBox(
2920.           width: 140,
2921.           child: pw.Text(
2922.             label,
2923.             style: pw.TextStyle(
2924.               font: font,
2925.               fontSize: 12,
2926.               fontWeight: pw.FontWeight.bold,
2927.               color: PdfColors.grey600,
2928.             ),
2929.           ),
2930.         pw.Expanded(
2931.           child: pw.Text(
2932.             value,
2933.             style: pw.TextStyle(font: font, fontSize:
2934.               12),
2935.           ),
2936.         ),
2937.       ],
2938.     );
2939.   }
2940.   @override
2941.   Widget build(BuildContext context) {
2942.     final Map<String, dynamic>? args =
2943.       ModalRoute.of(context)?.settings.arguments
2944.         as Map<String, dynamic>?;
2945.     final DateTime? date = args?['date'] as
2946.       DateTime?;
2947.     final String? time = args?['time'] as String?;
2948.     final String? doctor = args?['doctor'] as String?;
2949.     final String? department = args?['department']
2950.       as String?;
2951.     final String? appointmentType =
2952.       args?['appointmentType'] as String?;
2953.     final String? reason = args?['reason'] as
2954.       String?;
2955.     final String? username = args?['username'] as
2956.       String?;
2957.     final id = date != null
2958.       ? (100000 + (date.millisecondsSinceEpoch %
2959.         900000)).toString()
2960.       : 'N/A';
2961.     return Scaffold(
2962.       appBar: AppBar(
2963.         leading: IconButton(
2964.           icon: Icon(Icons.arrow_back, color:
2965.             Colors.indigo[700]),
2966.           onPressed: () {
2967.             Navigator.pop(context);
2968.             final tabController =
2969.               DefaultTabController.of(context);
2970.             if (tabController != null) {
2971.               tabController.animateTo(1);
2972.             }
2973.           },
2974.         ),
2975.         title: Text(
2976.           'Appointment Receipt',
2977.           style: GoogleFonts.poppins(
2978.             color: Colors.indigo[900],
```

GCARE CONNECT main.dart

```
2972.      fontSize:
MediaQuery.of(context).size.width < 600 ? 20 : 24,
2973.      fontWeight: FontWeight.w600,
2974.    ),
2975.  ),
2976.  backgroundColor: Colors.white,
2977.  elevation: 0,
2978.  shadowColor: Colors.black12,
2979. ),
2980. body: SingleChildScrollView(
2981.   child: LayoutBuilder(
2982.     builder: (context, constraints) {
2983.       final isMobile = constraints.maxWidth <
600;
2984.       return Container(
2985.         color: Color(0xFF8FAFC),
2986.         padding: EdgeInsets.symmetric(
2987.           vertical: isMobile ? 24 : 40,
2988.           horizontal: constraints.maxWidth *
0.05,
2989.         ),
2990.         child: Column(
2991.           children: [
2992.             SectionHeader(
2993.               title: 'Appointment Receipt',
2994.               subtitle:
2995.                 'Your appointment has been
successfully scheduled.',
2996.             ),
2997.             SizedBox(height: isMobile ? 12 : 24),
2998.             Card(
2999.               child: Padding(
3000.                 padding: EdgeInsets.all(isMobile ?
16 : 24),
3001.               child: Column(
3002.                 crossAxisAlignment:
CrossAxisAlignment.start,
3003.                 children: [
3004.                   Row(
3005.                     children: [
3006.                       Icon(
3007.                         Icons.check_circle,
3008.                         color: Colors.green,
3009.                         size: isMobile ? 24 : 28,
3010.                       ),
3011.                       SizedBox(width: 8),
3012.                       Text(
3013.                         'Appointment Confirmed',
3014.                         style: GoogleFonts.poppins(
3015.                           fontSize: isMobile ? 16 :
18,
3016.                           fontWeight:
FontWeight.w500,
3017.                         ),
3018.                     ),
3019.                   ],
3020.                 ),
3021.                 SizedBox(height: 12),
3022.                 Container(
3023.                   decoration: BoxDecoration(
3024.                     color:
Colors.indigo.withOpacity(0.05),
3025.                     borderRadius:
BorderRadius.circular(12),
3026.                     border: Border.all(
3027.                       color:
Colors.indigo.withOpacity(0.2),
3028.                     ),
3029.                   ),
3030.                   padding: EdgeInsets.all(12),
3031.                   width: double.infinity,
3032.                   child: Column(
3033.                     crossAxisAlignment:
CrossAxisAlignment.start,
3034.                     children: [
3035.                       Text(
3036.                         'Appointment Details',
3037.                         style:
GoogleFonts.poppins(
3038.                           fontSize: isMobile ? 14 :
16,
3039.                           fontWeight:
FontWeight.w500,
3040.                         ),
3041.                       ),
3042.                       SizedBox(height: 8),
3043.                       _summaryRow(
3044.                         'Appointment ID:',
3045.                         '#$id',
3046.                         isMobile,
3047.                       ),
3048.                       _summaryRow(
3049.                         'Patient:',
3050.                         username ?? 'Guest',
3051.                         isMobile,
3052.                       ),
3053.                       _summaryRow(
3054.                         'Date:',
3055.                         date != null
3056.                           ? DateFormat('MMMM
d, yyyy').format(date)
3057.                           : 'N/A',
3058.                         isMobile,
3059.                       ),
3060.                       _summaryRow('Time:', time
?? 'N/A', isMobile),
3061.                       _summaryRow(
3062.                         'Doctor:',
3063.                         doctor ?? 'N/A',
3064.                         isMobile,
3065.                       ),
3066.                       _summaryRow(
3067.                         'Department:',
3068.                         department ?? 'N/A',
3069.                         isMobile,
3070.                       ),
3071.                       _summaryRow(
3072.                         'Type:',
3073.                         appointmentType ?? 'N/A',
3074.                         isMobile,
3075.                       ),
3076.                       _summaryRow(
3077.                         'Reason:',
3078.                         reason?.isEmpty ?? true
3079.                           ? 'Not specified'
3080.                           : reason!,
3081.                         isMobile,
3082.                       ),
3083.                     ],
3084.                   ),
3085.                 ],

```

GCARE CONNECT main.dart

```
3086.      SizedBox(height: 12),
3087.      Container(
3088.        decoration: BoxDecoration(
3089.          border: Border(
3090.            left: BorderSide(
3091.              color: Colors.indigo[700]!,
3092.              width: 3,
3093.            ),
3094.          ),
3095.        ),
3096.        padding: EdgeInsets.only(left:
3097.          12),
3098.        child: Column(
3099.          crossAxisAlignment:
3100.            CrossAxisAlignment.start,
3101.          children: [
3102.            Text(
3103.              'Confirmation',
3104.              style:
3105.                GoogleFonts.poppins(
3106.                  fontSize: isMobile ? 14 :
3107.                    16,
3108.                  fontWeight:
3109.                    FontWeight.w500,
3110.                ),
3111.            ),
3112.            SizedBox(height: 8),
3113.            Text(
3114.              'A confirmation email will
3115.              be sent to your registered email address.',
3116.              style:
3117.                GoogleFonts.poppins(
3118.                  fontSize: isMobile ? 12 :
3119.                    14,
3120.                ),
3121.            ),
3122.            SizedBox(height: 8),
3123.            Text(
3124.              'Location: J. Yulo Avenue,
3125.              Brgy. Canlubang',
3126.              style:
3127.                GoogleFonts.poppins(
3128.                  fontSize: isMobile ? 12 :
3129.                    14,
3130.                ),
3131.            ),
3132.            Row(
3133.              mainAxisAlignment:
3134.                MainAxisAlignment.end,
3135.              children: [
3136.                OutlinedButton(
3137.                  onPressed: () {
3138.                    Navigator.pop(context);
3139.                    final tabController =
3140.                      DefaultTabController.of(
3141.                        context,
3142.                      );
3143.                    if (tabController != null) {
3144.                      child: Text('Back to
3145.                        Appointments'),
3146.                    ),
3147.                    SizedBox(width: 8),
3148.                    ElevatedButton.icon(
3149.                      onPressed: () =>
3150.                        context,
3151.                        id: id,
3152.                        username: username,
3153.                        date: date,
3154.                        time: time,
3155.                        doctor: doctor,
3156.                        department: department,
3157.                        appointmentType:
3158.                          reason: reason,
3159.                        ),
3160.                        icon: Icon(
3161.                          Icons.download,
3162.                          size: isMobile ? 16 : 18,
3163.                        ),
3164.                        label: Text('Download
3165.                          Receipt'),
3166.                        ElevatedButton.styleFrom(
3167.                          Colors.indigo[700],
3168.                          Colors.white,
3169.                          padding:
3170.                            EdgeInsets.symmetric(
3171.                              horizontal: isMobile ? 12
3172.                                : 16,
3173.                              vertical: isMobile ? 8 :
3174.                                10,
3175.                            ),
3176.                        ),
3177.                        ),
3178.                        ),
3179.                        ),
3180.                        ),
3181.                        ),
3182.                        );
3183.                      },
3184.                    ),
3185.                  ),
3186.                );
3187.              }
3188.            ),
3189.            Widget _summaryRow(String label, String value,
3190.              bool isMobile) {
```

GCARE CONNECT main.dart

```
3190.         return Padding(
3191.           padding: EdgeInsets.symmetric(vertical: 4),
3192.           child: Row(
3193.             crossAxisAlignment:
3194.               CrossAxisAlignment.start,
3195.             children: [
3196.               SizedBox(
3197.                 width: isMobile ? 120 : 140,
3198.                 child: Text(
3199.                   label,
3200.                   style: GoogleFonts.poppins(
3201.                     fontSize: isMobile ? 12 : 14,
3202.                     fontWeight: FontWeight.w500,
3203.                     color: Colors.grey[600],
3204.                   ),
3205.                 ),
3206.               Expanded(
3207.                 child: Text(
3208.                   value,
3209.                   style: GoogleFonts.poppins(
3210.                     fontSize: isMobile ? 12 : 14,
3211.                     fontWeight: FontWeight.w400,
3212.                   ),
3213.                 ),
3214.               ),
3215.             ],
3216.           ),
3217.         );
3218.       }
3219.     }
3220.
3221.     class AppointmentHistory extends StatelessWidget
3222.     {
3223.       final bool isMobile;
3224.       AppointmentHistory({required this.isMobile}); //
3225.       Fixed constructor
3226.       @override
3227.       Widget build(BuildContext context) {
3228.         return FutureBuilder<List<Map<String,
3229.           dynamic>>>(
3230.           future:
3231.             Provider.of<AppointmentService>(context).loadAppointment
3232.             s(),
3233.           builder: (context, snapshot) {
3234.             if (snapshot.connectionState ==
3235.               ConnectionState.waiting) {
3236.               return Center(child:
3237.                 CircularProgressIndicator());
3238.             }
3239.             if (snapshot.hasError) {
3240.               return Center(child: Text('Error loading
3241.                 appointments'));
3242.             }
3243.             final appointments = snapshot.data ?? [];
3244.             if (appointments.isEmpty) {
3245.               return Center(
3246.                 child: Text(
3247.                   'No appointments found.',
3248.                   style: GoogleFonts.poppins(fontSize:
3249.                     isMobile ? 16 : 18),
3250.                 ),
3251.               );
3252.             }
3253.           }
3254.         );
3255.       }
3256.     }
3257.
3258.     return ListView.separated(
3259.       shrinkWrap: true,
3260.       physics: NeverScrollableScrollPhysics(),
3261.       itemCount: appointments.length,
3262.       separatorBuilder: (context, index) =>
3263.         SizedBox(height: 12),
3264.       itemBuilder: (context, index) {
3265.         final appointment = appointments[index];
3266.         return Card(
3267.           child: Padding(
3268.             padding: EdgeInsets.all(isMobile ? 12 :
3269.               16),
3270.             child: Row(
3271.               crossAxisAlignment:
3272.                 CrossAxisAlignment.start,
3273.               children: [
3274.                 CircleAvatar(
3275.                   backgroundImage:
3276.                     NetworkImage(appointment['avatar']),
3277.                   radius: isMobile ? 24 : 28,
3278.                 ),
3279.                 SizedBox(width: 12),
3280.                 Expanded(
3281.                   child: Column(
3282.                     crossAxisAlignment:
3283.                       CrossAxisAlignment.start,
3284.                     children: [
3285.                       Text(
3286.                         appointment['doctor'] ?? 'N/A',
3287.                         style: GoogleFonts.poppins(
3288.                           fontSize: isMobile ? 16 : 18,
3289.                           fontWeight:
3290.                             FontWeight.w500,
3291.                         ),
3292.                       ),
3293.                       ),
3294.                     ),
3295.                   ),
3296.                 ),
3297.                 ),
3298.                 ),
3299.                 ),
3300.                 ),
3301.                 ),
3302.                 ),
3303.                 ),
3304.                 ),
3305.                 ),
3306.                 ),
3307.                 ),
3308.                 ),
3309.                 ),
3310.                 ),
3311.                 ),
3312.                 ),
3313.                 ),
3314.                 ),
3315.                 ),
3316.                 ),
3317.                 ),
3318.                 ),
3319.                 ),
3320.                 ),
3321.                 ),
3322.                 ),
3323.                 ),
3324.                 ),
3325.                 ),
3326.                 ),
3327.                 ),
3328.                 ),
3329.                 ),
3330.                 ),
3331.                 ),
3332.                 ),
3333.                 ),
3334.                 ),
3335.                 ),
3336.                 ),
3337.                 ),
3338.                 ),
3339.                 ),
3340.                 ),
3341.                 ),
3342.                 ),
3343.                 ),
3344.                 ),
3345.                 ),
3346.                 ),
3347.                 ),
3348.                 ),
3349.                 ),
3350.                 ),
3351.                 ),
3352.                 ),
3353.                 ),
3354.                 ),
3355.                 ),
3356.                 ),
3357.                 ),
3358.                 ),
3359.                 ),
3360.                 ),
3361.                 ),
3362.                 ),
3363.                 ),
3364.                 ),
3365.                 ),
3366.                 ),
3367.                 ),
3368.                 ),
3369.                 ),
3370.                 ),
3371.                 ),
3372.                 ),
3373.                 ),
3374.                 ),
3375.                 ),
3376.                 ),
3377.                 ),
3378.                 ),
3379.                 ),
3380.                 ),
3381.                 ),
3382.                 ),
3383.                 ),
3384.                 ),
3385.                 ),
3386.                 ),
3387.                 ),
3388.                 ),
3389.                 ),
3390.                 ),
3391.                 ),
3392.                 ),
3393.                 ),
3394.                 ),
3395.                 ),
3396.                 ),
3397.                 ),
3398.                 ),
3399.                 ),
3400.                 ),
3401.                 ),
3402.                 ),
3403.                 ),
3404.                 ),
3405.                 ),
3406.                 ),
3407.                 ),
3408.                 ),
3409.                 ),
3410.                 ),
3411.                 ),
3412.                 ),
3413.                 ),
3414.                 ),
3415.                 ),
3416.                 ),
3417.                 ),
3418.                 ),
3419.                 ),
3420.                 ),
3421.                 ),
3422.                 ),
3423.                 ),
3424.                 ),
3425.                 ),
3426.                 ),
3427.                 ),
3428.                 ),
3429.                 ),
3430.                 ),
3431.                 ),
3432.                 ),
3433.                 ),
3434.                 ),
3435.                 ),
3436.                 ),
3437.                 ),
3438.                 ),
3439.                 ),
3440.                 ),
3441.                 ),
3442.                 ),
3443.                 ),
3444.                 ),
3445.                 ),
3446.                 ),
3447.                 ),
3448.                 ),
3449.                 ),
3450.                 ),
3451.                 ),
3452.                 ),
3453.                 ),
3454.                 ),
3455.                 ),
3456.                 ),
3457.                 ),
3458.                 ),
3459.                 ),
3460.                 ),
3461.                 ),
3462.                 ),
3463.                 ),
3464.                 ),
3465.                 ),
3466.                 ),
3467.                 ),
3468.                 ),
3469.                 ),
3470.                 ),
3471.                 ),
3472.                 ),
3473.                 ),
3474.                 ),
3475.                 ),
3476.                 ),
3477.                 ),
3478.                 ),
3479.                 ),
3480.                 ),
3481.                 ),
3482.                 ),
3483.                 ),
3484.                 ),
3485.                 ),
3486.                 ),
3487.                 ),
3488.                 ),
3489.                 ),
3490.                 ),
3491.                 ),
3492.                 ),
3493.                 ),
3494.                 ),
3495.                 ),
3496.                 ),
3497.                 ),
3498.                 ),
3499.                 ),
3500.                 ),
3501.                 ),
3502.                 ),
3503.                 ),
3504.                 ),
3505.                 ),
3506.                 ),
3507.                 ),
3508.                 ),
3509.                 ),
3510.                 ),
3511.                 ),
3512.                 ),
3513.                 ),
3514.                 ),
3515.                 ),
3516.                 ),
3517.                 ),
3518.                 ),
3519.                 ),
3520.                 ),
3521.                 ),
3522.                 ),
3523.                 ),
3524.                 ),
3525.                 ),
3526.                 ),
3527.                 ),
3528.                 ),
3529.                 ),
3530.                 ),
3531.                 ),
3532.                 ),
3533.                 ),
3534.                 ),
3535.                 ),
3536.                 ),
3537.                 ),
3538.                 ),
3539.                 ),
3540.                 ),
3541.                 ),
3542.                 ),
3543.                 ),
3544.                 ),
3545.                 ),
3546.                 ),
3547.                 ),
3548.                 ),
3549.                 ),
3550.                 ),
3551.                 ),
3552.                 ),
3553.                 ),
3554.                 ),
3555.                 ),
3556.                 ),
3557.                 ),
3558.                 ),
3559.                 ),
3560.                 ),
3561.                 ),
3562.                 ),
3563.                 ),
3564.                 ),
3565.                 ),
3566.                 ),
3567.                 ),
3568.                 ),
3569.                 ),
3570.                 ),
3571.                 ),
3572.                 ),
3573.                 ),
3574.                 ),
3575.                 ),
3576.                 ),
3577.                 ),
3578.                 ),
3579.                 ),
3580.                 ),
3581.                 ),
3582.                 ),
3583.                 ),
3584.                 ),
3585.                 ),
3586.                 ),
3587.                 ),
3588.                 ),
3589.                 ),
3590.                 ),
3591.                 ),
3592.                 ),
3593.                 ),
3594.                 ),
3595.                 ),
3596.                 ),
3597.                 ),
3598.                 ),
3599.                 ),
3600.                 ),
3601.                 ),
3602.                 ),
3603.                 ),
3604.                 ),
3605.                 ),
3606.                 ),
3607.                 ),
3608.                 ),
3609.                 ),
3610.                 ),
3611.                 ),
3612.                 ),
3613.                 ),
3614.                 ),
3615.                 ),
3616.                 ),
3617.                 ),
3618.                 ),
3619.                 ),
3620.                 ),
3621.                 ),
3622.                 ),
3623.                 ),
3624.                 ),
3625.                 ),
3626.                 ),
3627.                 ),
3628.                 ),
3629.                 ),
3630.                 ),
3631.                 ),
3632.                 ),
3633.                 ),
3634.                 ),
3635.                 ),
3636.                 ),
3637.                 ),
3638.                 ),
3639.                 ),
3640.                 ),
3641.                 ),
3642.                 ),
3643.                 ),
3644.                 ),
3645.                 ),
3646.                 ),
3647.                 ),
3648.                 ),
3649.                 ),
3650.                 ),
3651.                 ),
3652.                 ),
3653.                 ),
3654.                 ),
3655.                 ),
3656.                 ),
3657.                 ),
3658.                 ),
3659.                 ),
3660.                 ),
3661.                 ),
3662.                 ),
3663.                 ),
3664.                 ),
3665.                 ),
3666.                 ),
3667.                 ),
3668.                 ),
3669.                 ),
3670.                 ),
3671.                 ),
3672.                 ),
3673.                 ),
3674.                 ),
3675.                 ),
3676.                 ),
3677.                 ),
3678.                 ),
3679.                 ),
3680.                 ),
3681.                 ),
3682.                 ),
3683.                 ),
3684.                 ),
3685.                 ),
3686.                 ),
3687.                 ),
3688.                 ),
3689.                 ),
3690.                 ),
3691.                 ),
3692.                 ),
3693.                 ),
3694.                 ),
3695.                 ),
3696.                 ),
3697.                 ),
3698.                 ),
3699.                 ),
3700.                 ),
3701.                 ),
3702.                 ),
3703.                 ),
3704.                 ),
3705.                 ),
3706.                 ),
3707.                 ),
3708.                 ),
3709.                 ),
3710.                 ),
3711.                 ),
3712.                 ),
3713.                 ),
3714.                 ),
3715.                 ),
3716.                 ),
3717.                 ),
3718.                 ),
3719.                 ),
3720.                 ),
3721.                 ),
3722.                 ),
3723.                 ),
3724.                 ),
3725.                 ),
3726.                 ),
3727.                 ),
3728.                 ),
3729.                 ),
3730.                 ),
3731.                 ),
3732.                 ),
3733.                 ),
3734.                 ),
3735.                 ),
3736.                 ),
3737.                 ),
3738.                 ),
3739.                 ),
3740.                 ),
3741.                 ),
3742.                 ),
3743.                 ),
3744.                 ),
3745.                 ),
3746.                 ),
3747.                 ),
3748.                 ),
3749.                 ),
3750.                 ),
3751.                 ),
3752.                 ),
3753.                 ),
3754.                 ),
3755.                 ),
3756.                 ),
3757.                 ),
3758.                 ),
3759.                 ),
3760.                 ),
3761.                 ),
3762.                 ),
3763.                 ),
3764.                 ),
3765.                 ),
3766.                 ),
3767.                 ),
3768.                 ),
3769.                 ),
3770.                 ),
3771.                 ),
3772.                 ),
3773.                 ),
3774.                 ),
3775.                 ),
3776.                 ),
3777.                 ),
3778.                 ),
3779.                 ),
3780.                 ),
3781.                 ),
3782.                 ),
3783.                 ),
3784.                 ),
3785.                 ),
3786.                 ),
3787.                 ),
3788.                 ),
3789.                 ),
3790.                 ),
3791.                 ),
3792.                 ),
3793.                 ),
3794.                 ),
3795.                 ),
3796.                 ),
3797.                 ),
3798.                 ),
3799.                 ),
3800.                 ),
3801.                 ),
3802.                 ),
3803.                 ),
3804.                 ),
3805.                 ),
3806.                 ),
3807.                 ),
3808.                 ),
3809.                 ),
3810.                 ),
3811.                 ),
3812.                 ),
3813.                 ),
3814.                 ),
3815.                 ),
3816.                 ),
3817.                 ),
3818.                 ),
3819.                 ),
3820.                 ),
3821.                 ),
3822.                 ),
3823.                 ),
3824.                 ),
3825.                 ),
3826.                 ),
3827.                 ),
3828.                 ),
3829.                 ),
3830.                 ),
3831.                 ),
3832.                 ),
3833.                 ),
3834.                 ),
3835.                 ),
3836.                 ),
3837.                 ),
3838.                 ),
3839.                 ),
3840.                 ),
3841.                 ),
3842.                 ),
3843.                 ),
3844.                 ),
3845.                 ),
3846.                 ),
3847.                 ),
3848.                 ),
3849.                 ),
3850.                 ),
3851.                 ),
3852.                 ),
3853.                 ),
3854.                 ),
3855.                 ),
3856.                 ),
3857.                 ),
3858.                 ),
3859.                 ),
3860.                 ),
3861.                 ),
3862.                 ),
3863.                 ),
3864.                 ),
3865.                 ),
3866.                 ),
3867.                 ),
3868.                 ),
3869.                 ),
3870.                 ),
3871.                 ),
3872.                 ),
3873.                 ),
3874.                 ),
3875.                 ),
3876.                 ),
3877.                 ),
3878.                 ),
3879.                 ),
3880.                 ),
3881.                 ),
3882.                 ),
3883.                 ),
3884.                 ),
3885.                 ),
3886.                 ),
3887.                 ),
3888.                 ),
3889.                 ),
3890.                 ),
3891.                 ),
3892.                 ),
3893.                 ),
3894.                 ),
3895.                 ),
3896.                 ),
3897.                 ),
3898.                 ),
3899.                 ),
3900.                 ),
3901.                 ),
3902.                 ),
3903.                 ),
3904.                 ),
3905.                 ),
3906.                 ),
3907.                 ),
3908.                 ),
3909.                 ),
3910.                 ),
3911.                 ),
3912.                 ),
3913.                 ),
3914.                 ),
3915.                 ),
3916.                 ),
3917.                 ),
3918.                 ),
3919.                 ),
3920.                 ),
3921.                 ),
3922.                 ),
3923.                 ),
3924.                 ),
3925.                 ),
3926.                 ),
3927.                 ),
3928.                 ),
3929.                 ),
3930.                 ),
3931.                 ),
3932.                 ),
3933.                 ),
3934.                 ),
3935.                 ),
3936.                 ),
3937.                 ),
3938.                 ),
3939.                 ),
3940.                 ),
3941.                 ),
3942.                 ),
3943.                 ),
3944.                 ),
3945.                 ),
3946.                 ),
3947.                 ),
3948.                 ),
3949.                 ),
3950.                 ),
3951.                 ),
3952.                 ),
3953.                 ),
3954.                 ),
3955.                 ),
3956.                 ),
3957.                 ),
3958.                 ),
3959.                 ),
3960.                 ),
3961.                 ),
3962.                 ),
3963.                 ),
3964.                 ),
3965.                 ),
3966.                 ),
3967.                 ),
3968.                 ),
3969.                 ),
3970.                 ),
3971.                 ),
3972.                 ),
3973.                 ),
3974.                 ),
3975.                 ),
3976.                 ),
3977.                 ),
3978.                 ),
3979.                 ),
3980.                 ),
3981.                 ),
3982.                 ),
3983.                 ),
3984.                 ),
3985.                 ),
3986.                 ),
3987.                 ),
3988.                 ),
3989.                 ),
3990.                 ),
3991.                 ),
3992.                 ),
3993.                 ),
3994.                 ),
3995.                 ),
3996.                 ),
3997.                 ),
3998.                 ),
3999.                 ),
4000.                 ),
4001.                 ),
4002.                 ),
4003.                 ),
4004.                 ),
4005.                 ),
4006.                 ),
4007.                 ),
4008.                 ),
4009.                 ),
4010.                 ),
4011.                 ),
4012.                 ),
4013.                 ),
4014.                 ),
4015.                 ),
4016.                 ),
4017.                 ),
4018.                 ),
4019.                 ),
4020.                 ),
4021.                 ),
4022.                 ),
4023.                 ),
4024.                 ),
4025.                 ),
4026.                 ),
4027.                 ),
4028.                 ),
4029.                 ),
4030.                 ),
4031.                 ),
4032.                 ),
4033.                 ),
4034.                 ),
4035.                 ),
4036.                 ),
4037.                 ),
4038.                 ),
4039.                 ),
4040.                 ),
4041.                 ),
4042.                 ),
4043.                 ),
4044.                 ),
4045.                 ),
4046.                 ),
4047.                 ),
4048.                 ),
4049.                 ),
4050.                 ),
4051.                 ),
4052.                 ),
4053.                 ),
4054.                 ),
4055.                 ),
4056.                 ),
4057.                 ),
4058.                 ),
4059.                 ),
4060.                 ),
4061.                 ),
4062.                 ),
4063.                 ),
4064.                 ),
4065.                 ),
4066.                 ),
4067.                 ),
4068.                 ),
4069.                 ),
4070.                 ),
4071.                 ),
4072.                 ),
4073.                 ),
4074.                 ),
4075.                 ),
4076.                 ),
4077.                 ),
4078.                 ),
4079.                 ),
4080.                 ),
4081.                 ),
4082.                 ),
4083.                 ),
4084.                 ),
4085.                 ),
4086.                 ),
4087.                 ),
4088.                 ),
4089.                 ),
4090.                 ),
4091.                 ),
4092.                 ),
4093.                 ),
4094.                 ),
4095.                 ),
4096.                 ),
4097.                 ),
4098.                 ),
4099.                 ),
4100.                 ),
4101.                 ),
4102.                 ),
4103.                 ),
4104.                 ),
4105.                 ),
4106.                 ),
4107.                 ),
4108.                 ),
4109.                 ),
4110.                 ),
4111.                 ),
4112.                 ),
4113.                 ),
4114.                 ),
4115.                 ),
4116.                 ),
4117.                 ),
4118.                 ),
4119.                 ),
4120.                 ),
4121.                 ),
4122.                 ),
4123.                 ),
4124.                 ),
4125.                 ),
4126.                 ),
4127.                 ),
4128.                 ),
4129.                 ),
4130.                 ),
4131.                 ),
4132.                 ),
4133.                 ),
4134.                 ),
4135.                 ),
4136.                 ),
4137.                 ),
4138.                 ),
4139.                 ),
4140.                 ),
4141.                 ),
4142.                 ),
4143.                 ),
4144.                 ),
4145.                 ),
4146.                 ),
4147.                 ),
4148.                 ),
4149.                 ),
4150.                 ),
4151.                 ),
4152.                 ),
4153.                 ),
4154.                 ),
4155.                 ),
4156.                 ),
4157.                 ),
4158.                 ),
4159.                 ),
4160.                 ),
4161.                 ),
4162.                 ),
4163.                 ),
4164.                 ),
4165.                 ),
4166.                 ),
4167.                 ),
4168.                 ),
4169.                 ),
4170.                 ),
4171.                 ),
4172.                 ),
4173.                 ),
4174.                 ),
4175.                 ),
4176.                 ),
4177.                 ),
4178.                 ),
4179.                 ),
4180.                 ),
4181.                 ),
4182.                 ),
4183.                 ),
4184.                 ),
4185.                 ),
4186.                 ),
4187.                 ),
4188.                 ),
4189.                 ),
4190.                 ),
4191.                 ),
4192.                 ),
4193.                 ),
4194.                 ),
4195.                 ),
4196.                 ),
4197.                 ),
4198.                 ),
4199.                 ),
4200.                 ),
4201.                 ),
4202.                 ),
4203.                 ),
4204.                 ),
4205.                 ),
4206.                 ),
4207.                 ),
4208.                 ),
4209.                 ),
4210.                 ),
4211.                 ),
4212.                 ),
4213.                 ),
4214.                 ),
4215.                 ),
4216.                 ),
4217.                 ),
4218.                 ),
4219.                 ),
4220.                 ),
4221.                 ),
4222.                 ),
4223.                 ),
4224.                 ),
4225.                 ),
4226.                 ),
4227.                 ),
4228.                 ),
4229.                 ),
4230.                 ),
4231.                 ),
4232.                 ),
4233.                 ),
4234.                 ),
4235.                 ),
4236.                 ),
4237.                 ),
4238.                 ),
4239.                 ),
4240.                 ),
4241.                 ),
4242.                 ),
4243.                 ),
4244.                 ),
4245.                 ),
4246.                 ),
4247.                 ),
4248.                 ),
4249.                 ),
4250.                 ),
4251.                 ),
4252.                 ),
4253.                 ),
4254.                 ),
4255.                 ),
4256.                 ),
4257.                 ),
4258.                 ),
4259.                 ),
4260.                 ),
4261.                 ),
4262.                 ),
4263.                 ),
4264.                 ),
4265.                 ),
4266.                 ),
4267.                 ),
4268.                 ),
4269.                 ),
4270.                 ),
4271.                 ),
4272.                 ),
4273.                 ),
4274.                 ),
4275.                 ),
4276.                 ),
4277.                 ),
4278.                 ),
4279.                 ),
4280.                 ),
4281.                 ),
4282.                 ),
4283.                 ),
4284.                 ),
4285.                 ),
4286.                 ),
4287.                 ),
4288.                 ),
4289.                 ),
4290.                 ),
4291.                 ),
4292.                 ),
4293.                 ),
4294.                 ),
4295.                 ),
4296.                 ),
4297.                 ),
4298.                 ),
4299.                 ),
4300.                 ),
4301.                 ),
4302.                 ),
4303.                 ),
4304.                 ),
4305.                 ),
4306.                 ),
4307.                 ),
4308.                 ),
4309.                 ),
4310.                 ),
4311.                 ),
4312.                 ),
4313.                 ),
4314.                 ),
4315.                 ),
4316.                 ),
4317.                 ),
4318.                 ),
4319.                 ),
4320.                 ),
4321.                 ),
4322.                 ),
4323.                 ),
4324.                 ),
4325.                 ),
4326.                 ),
4327.                 ),
4328.                 ),
4329.                 ),
4330.                 ),
4331.                 ),
4332.                 ),
4333.                 ),
4334.                 ),
4335.                 ),
4336.                 ),
4337.                 ),
4338.                 ),
4339.                 ),
4340.                 ),
4341.                 ),
4342.                 ),
4343.                 ),
4344.                 ),
4345.                 ),
4346.                 ),
4347.                 ),
4348.                 ),
4349.                 ),
4350.                 ),
4351.                 ),
4352.                 ),
4353.                 ),
4354.                 ),
4355.                 ),
4356.                 ),
4357.                 ),
4358.                 ),
4359.                 ),
4360.                 ),
4361.                 ),
4362.                 ),
4363.                 ),
4364.                 ),
4365.                 ),
4366.                 ),
4367.                 ),
4368.                 ),
4369.                 ),
4370.                 ),
4371.                 ),
4372.                 ),
4373.                 ),
4374.                 ),
4375.                 ),
4376.                 ),
4377.                 ),
4378.                 ),
4379.                 ),
4380.                 ),
4381.                 ),
4382.                 ),
4383.                 ),
4384.                 ),
4385.                 ),
4386.                 ),
4387.                 ),
4388.                 ),
4389.                 ),
4390.                 ),
4391.                 ),
4392.                 ),
4393.                 ),
4394.                 ),
4395.                 ),
4396.                 ),
4397.                 ),
4398.                 ),
4399.                 ),
4400.                 ),
4401.                 ),
4402.                 ),
4403.                 ),
4404.                 ),
4405.                 ),
4406.                 ),
4407.                 ),
4408.                 ),
4409.                 ),
4410.                 ),
4411.                 ),
4412.                 ),
4413.                 ),
4414.                 ),
4415.                 ),
4416.                 ),
4417.                 ),
4418.                 ),
4419.                 ),
4420.                 ),
4421.                 ),
4422.                 ),
4423.                 ),
4424.                 ),
4425.                 ),
4426.                 ),
4427.                 ),
4428.                 ),
4429.                 ),
4430.                 ),
4431.                 ),
4432.                 ),
4433.                 ),
4434.                 ),
4435.                 ),
4436.                 ),
4437.                 ),
4438.                 ),
4439.                 ),
4440.                 ),
4441.                 ),
4442.                 ),
4443.                 ),
4444.                 ),
4445.                 ),
4446.                 ),
4447.                 ),
4448.                 ),
4449.                 ),
4450.                 ),
4451.                 ),
4452.                 ),
4453.                 ),
4454.                 ),
4455.                 ),
4456.                 ),
4457.                 ),
4458.                 ),
4459.                 ),
4460.                 ),
4461.                 ),
4462.                 ),
4463.                 ),
4464.                 ),
4465.                 ),
4466.                 ),
4467.                 ),
4468.                 ),
4469.                 ),
4470.                 ),
4471.                 ),
4472.                 ),
4473.                 ),
4474.                 ),
4475.                 ),
4476.                 ),
4477.                 ),
4478.                 ),
4479.                 ),
4480.                 ),
4481.                 ),
4482.                 ),
4483.                 ),
4484.                 ),
4485.                 ),
4486.                 ),
4487.                 ),
4488.                 ),
4489.                 ),
4490.                 ),
4491.                 ),
4492.                 ),
4493.                 ),
4494.                 ),
4495.                 ),
4496.                 ),
4497.                 ),
4498.                 ),
4499.                 ),
4500.                 ),
4501.                 ),
4502.                 ),
4503.                 ),
4504.                 ),
4505.                 ),
4506.                 ),
4507.                 ),
4508.                 ),
4509.                 ),
4510.                 ),
4511.                 ),
4512.                 ),
4513.                 ),
4514.                 ),
4515.                 ),
4516.                 ),
4517.                 ),
4518.                 ),
4519.                 ),
4520.                 ),
4521.                 ),
4522.                 ),
4523.                 ),
4524.                 ),
4525.                 ),
4526.                 ),
4527.                 ),
4528.                 ),
4529.                 ),
4530.                 ),
4531.                 ),
4532.                 ),
4533.                 ),
4534.                 ),
4535.                 ),
4536.                 ),
4537.                 ),
4538.                 ),
4539.                 ),
4540.                 ),
4541.                 ),
4542.                 ),
4543.                 ),
4544.                 ),
4545.                 ),
4546.                 ),
4547.                 ),
4548.                 ),
4549.                 ),
4550.                 ),
4551.                 ),
4552.                 ),
4553.                 ),
4554.                 ),
4555.                 ),
4556.                 ),
4557.                 ),
4558.                 ),
4559.                 ),
4560.                 ),
4561.                 ),
4562.                 ),
4563.                 ),
4564.                 ),
4565.                 ),
4566.                 ),
4567.                 ),
4568.                 ),
4569.                 ),
4570.                 ),
4571.                 ),
4572.                 ),
4573.                 ),
4574.                 ),
4575.                 ),
4576.                 ),
4577.                 ),
4578.                 ),
4579.                 ),
4580.                 ),
4581.                 ),
4582.                 ),
4583.                 ),
4584.                 ),
4585.                 ),
4586.                 ),
4587.                 ),
4588.                 ),
4589.                 ),
4590.                 ),
4591.                 ),
4592.                 ),
4593.                 ),
4594.                 ),
4595.                 ),
4596.                 ),
4597.                 ),
4598.                 ),
4599.                 ),
4600.                 ),
4601.                 ),
4602.                 ),
4603.                 ),
4604.                 ),
4605.                 ),
4606.                 ),
4607.                 ),
4608.                 ),
4609.                 ),
4610.                 ),
4611.                 ),
4612.                 ),
4613.                 ),
4614.                 ),
4615.                 ),
4616.                 ),
4617.                 ),
4618.                 ),
4619.                 ),
4620.                 ),
4621.                 ),
4622.                 ),
4623.                 ),
4624.                 ),
4625.                 ),
4626.                 ),
4627.                 ),
4628.                 ),
4629.                 ),
4630.                 ),
4631.                 ),
4632.                 ),
4633.                 ),
4634.                 ),
4635.                 ),
4636.                 ),
4637.                 ),
4638.                 ),
4639.                 ),
4640.                 ),
4641.                 ),
4642.                 ),
4643.                 ),
4644.                 ),
4645.                 ),
4646.                 ),
4647.                 ),
4648.                 ),
4649.                 ),
4650.                 ),
4651.                 ),
4652.                 ),
4653.                 ),
4654.                 ),
4655.                 ),
4656.                 ),
4657.                 ),
4658.                 ),
4659.                 ),
4660.                 ),
4661.                 ),
4662.                 ),
4663.                 ),
4664.                 ),
4665.                 ),
4666.                 ),
4667.                 ),
4668.                 ),
4669.                 ),
4670.                 ),
4671.                 ),
4672.                 ),
4673.                 ),
4674.                 ),
4675.                 ),
4676.                 ),
4677.                 ),
4678.                 ),
4679.                 ),
4680.                 ),
4681.                 ),
4682.                 ),
4683.                 ),
4
```

GCARE CONNECT main.dart

```

3304.      color: appointment['status']
      == 'Upcoming'
3305.      ? Colors.green
3306.      : Colors.grey[600],
3307.    ),
3308.  ),
3309.  if
    (appointment['reason']?.isEmpty ?? false)
3310.    Column(
3311.      crossAxisAlignment:
        CrossAxisAlignment.start,
3312.      children: [
3313.        SizedBox(height: 8),
3314.        Text(
3315.          'Reason:
        ${appointment['reason']}',
3316.          style:
        GoogleFonts.poppins(
3317.            fontSize: isMobile ? 14 :
        16,
3318.            color: Colors.grey[600],
3319.          ),
3320.        ),
3321.      ],
3322.    ),
3323.  ],
3324. ),
3325. ),
3326. ],
3327. ),
3328. ),
3329. );
3330. },
3331. );
3332. },
3333. );
3334. }
3335. }
3336.
3337. class ContactSection extends StatelessWidget {
3338.   @override
3339.   Widget build(BuildContext context) {
3340.     return LayoutBuilder(
3341.       builder: (context, constraints) {
3342.         final isMobile = constraints.maxWidth < 600;
3343.         return Container(
3344.           color: Color(0xFFFF8FAFC),
3345.           padding: EdgeInsets.symmetric(
3346.             vertical: isMobile ? 24 : 40,
3347.             horizontal: constraints.maxWidth * 0.05,
3348.           ),
3349.           child: Column(
3350.             children: [
3351.               SectionHeader(
3352.                 title: 'Contact Us',
3353.                 subtitle: 'Get in touch with our team for
        any inquiries.',
3354.               ),
3355.               SizedBox(height: isMobile ? 12 : 24),
3356.               Card(
3357.                 child: Padding(
3358.                   padding: EdgeInsets.all(isMobile ? 16
        : 24),
3359.                   child: isMobile
3360.                     ? Column(
3361.
3362.
3363.
3364.
3365.
3366.
3367.
3368.
3369.
3370.
3371.
3372.
3373.
3374.
3375.
3376.
3377.
3378.
3379.
3380.
3381.
3382.
3383.
3384.
3385.
3386.
3387.
3388.
3389.
3390.
3391.
3392.
3393.
3394.
3395.
3396.
3397.
3398.
3399.
3400.
3401.
3402.
3403.
3404.
3405.
3406.
3407.
3408.
3409.
3410.
3411.
3412.
3413.
3414.

```

GCARE CONNECT main.dart

```

3415.         child: ContactItem(
3416.             icon: Icons.location_on,
3417.             title: 'Address',
3418.             content: 'J. Yulo Avenue,
           Brgy. Canlubang',
3419.             isMobile: isMobile,
3420.             ),
3421.         ),
3422.         SizedBox(width: 24),
3423.         Expanded(
3424.             child: ContactItem(
3425.                 icon: Icons.phone,
3426.                 title: 'Phone',
3427.                 content: '(049) 520-5626',
3428.                 isMobile: isMobile,
3429.                 onTap: () async {
3430.                     final url =
                       Uri.parse('tel:(049) 520-5626');
3431.                     if (await
                       canLaunchUrl(url)) {
3432.                         await launchUrl(url);
3433.                     } else {
3434.                         ScaffoldMessenger.of(context).showSnackBar(
3435.                             SnackBar(
3436.                                 content: Text('Cannot
                                   launch phone'),
3437.                                 ),
3438.                             );
3439.                     }
3440.                 },
3441.             ),
3442.         ),
3443.         SizedBox(width: 24),
3444.         Expanded(
3445.             child: ContactItem(
3446.                 icon: Icons.email,
3447.                 title: 'Email',
3448.                 content:
                       'gcmccanlubang@gmail.com',
3449.                 isMobile: isMobile,
3450.                 onTap: () async {
3451.                     final url = Uri.parse(
                       'mailto:gcmccanlubang@gmail.com',
3452.                     );
3453.                     if (await
                       canLaunchUrl(url)) {
3454.                         await launchUrl(url);
3455.                     } else {
3456.                         ScaffoldMessenger.of(context).showSnackBar(
3457.                             SnackBar(
3458.                                 content: Text('Cannot
                                   launch email'),
3459.                                 ),
3460.                             );
3461.                     }
3462.                 },
3463.             ),
3464.         ),
3465.         ),
3466.         ],
3467.     ),
3468.   ),
3469. ],
3470. ],

```

```

3471.         ),
3472.     );
3473.     },
3474.     );
3475.     }
3476. }
3477.
3478. class ContactItem extends StatelessWidget {
3479.     final IconData icon;
3480.     final String title;
3481.     final String content;
3482.     final bool isMobile;
3483.     final VoidCallback? onTap;
3484.
3485.     ContactItem({
3486.         required this.icon,
3487.         required this.title,
3488.         required this.content,
3489.         required this.isMobile,
3490.         this.onTap,
3491.     });
3492.
3493.     @override
3494.     Widget build(BuildContext context) {
3495.         return InkWell(
3496.             onTap: onTap,
3497.             child: Row(
3498.                 crossAxisAlignment:
3499.                     CrossAxisAlignment.start,
3500.                 children: [
3501.                     CircleAvatar(
3502.                         backgroundColor:
3503.                             Colors.indigo.withOpacity(0.1),
3504.                         child: Icon(icon, color:
3505.                             Colors.indigo[700]),
3506.                         radius: isMobile ? 20 : 24,
3507.                     ),
3508.                     SizedBox(width: 12),
3509.                     Column(
3510.                         crossAxisAlignment:
3511.                             CrossAxisAlignment.start,
3512.                         children: [
3513.                             Text(
3514.                                 title,
3515.                                 style: GoogleFonts.poppins(
3516.                                     fontSize: isMobile ? 14 : 16,
3517.                                     fontWeight: FontWeight.w500,
3518.                                 ),
3519.                             ),
3520.                             SizedBox(height: 4),
3521.                             Text(
3522.                                 content,
3523.                                 style: GoogleFonts.poppins(
3524.                                     fontSize: isMobile ? 12 : 14,
3525.                                     color: Colors.grey[600],
3526.                                 ),
3527.                             ),
3528.                         ],
3529.                     ),
3530.                 ],
3531.             ),
3532.         );
3533.     }
3534. }
3535.
3536. class FooterSection extends StatelessWidget {

```

GCARE CONNECT main.dart

3533.	@override	3592.	}
3534.	Widget build(BuildContext context) {	3593.	},
3535.	return LayoutBuilder(3594.),
3536.	builder: (context, constraints) {	3595.	FooterLink(
3537.	final isMobile = constraints.maxWidth < 600;	3596.	text: 'Feedback',
3538.	return Container(3597.	onTap: () {
3539.	color: Colors.indigo[900],	3598.	final tabController =
3540.	padding: EdgeInsets.symmetric(DefaultTabController.of(context);
3541.	vertical: isMobile ? 24 : 40,	3599.	if (tabController != null) {
3542.	horizontal: constraints.maxWidth * 0.05,	3600.	tabController.animateTo(2);
3543.),	3601.	(context as
3544.	child: isMobile		Element).markNeedsBuild();
3545.	? Column(3602.	}
3546.	crossAxisAlignment:	3603.	},
	CrossAxisAlignment.start,	3604.),
3547.	children: [3605.	FooterLink(
3548.	Row(3606.	text: 'Contact Us',
3549.	children: [3607.	onTap: () {
3550.	Icon(3608.	// TODO: Implement scroll to
3551.	Icons.favorite,		contact section
3552.	color: Colors.white,	3609.	ScaffoldMessenger.of(context).showSnackBar(
3553.	size: isMobile ? 24 : 28,	3610.	SnackBar(content: Text('Contact
3554.),		Us clicked!')),
3555.	SizedBox(width: 8),	3611.);
3556.	Text(3612.	},
3557.	'Global Care',	3613.),
3558.	style: GoogleFonts.poppins(3614.	SizedBox(height: 24),
3559.	color: Colors.white,	3615.	Text(
3560.	fontSize: isMobile ? 20 : 24,	3616.	'© 2025 Global Care Medical
3561.	fontWeight: FontWeight.w700,		Center. All rights reserved.',
3562.),	3617.	style: GoogleFonts.poppins(
3563.),	3618.	color:
3564.],		Colors.white.withOpacity(0.7),
3565.),	3619.	fontSize: isMobile ? 12 : 14,
3566.	SizedBox(height: 16),	3620.),
3567.	Text(3621.),
3568.	'Providing compassionate care and	3622.],
	cutting-edge medical services to our community.',	3623.)
3569.	style: GoogleFonts.poppins(3624.	: Column(
3570.	color:	3625.	children: [
	Colors.white.withOpacity(0.9),	3626.	Row(
3571.	fontSize: isMobile ? 12 : 14,	3627.	crossAxisAlignment:
3572.),		CrossAxisAlignment.start,
3573.),	3628.	children: [
3574.	SizedBox(height: 24),	3629.	Expanded(
3575.	FooterLink(3630.	flex: 2,
3576.	text: 'Home',	3631.	child: Column(
3577.	onTap: () {	3632.	crossAxisAlignment:
3578.	final tabController =		CrossAxisAlignment.start,
	DefaultTabController.of(context);	3633.	children: [
3579.	if (tabController != null) {	3634.	Row(
3580.	tabController.animateTo(0);	3635.	children: [
3581.	(context as	3636.	Icon(
	Element).markNeedsBuild();	3637.	Icons.favorite,
3582.	}	3638.	color: Colors.white,
3583.	},	3639.	size: 28,
3584.),	3640.),
3585.	FooterLink(3641.	SizedBox(width: 8),
3586.	text: 'Appointments',	3642.	Text(
3587.	onTap: () {	3643.	'Global Care',
3588.	final tabController =	3644.	style:
	DefaultTabController.of(context);		GoogleFonts.poppins(
3589.	if (tabController != null) {	3645.	color: Colors.white,
3590.	tabController.animateTo(1);	3646.	fontSize: 24,
3591.	(context as		
	Element).markNeedsBuild();		

GCARE CONNECT main.dart

```
3647.          fontWeight:
          FontWeight.w700,
3648.          ),
3649.          ),
3650.        ],
3651.      ),
3652.      SizedBox(height: 16),
3653.      Text(
3654.        'Providing compassionate
        care and cutting-edge medical services to our community.',
3655.        style: GoogleFonts.poppins(
3656.          color:
            Colors.white.withOpacity(0.9),
3657.          fontSize: 14,
3658.        ),
3659.      ),
3660.    ],
3661.  ),
3662. ),
3663.  SizedBox(width: 24),
3664.  Expanded(
3665.    child: Column(
3666.      crossAxisAlignment:
        CrossAxisAlignment.start,
3667.      children: [
3668.        Text(
3669.          'Quick Links',
3670.          style: GoogleFonts.poppins(
3671.            color: Colors.white,
3672.            fontSize: 16,
3673.            fontWeight:
              FontWeight.w500,
3674.          ),
3675.        ),
3676.        SizedBox(height: 12),
3677.        FooterLink(
3678.          text: 'Home',
3679.          onTap: () {
3680.            final tabController =
              DefaultTabController.of(
3681.                context,
3682.              );
3683.              if (tabController != null) {
3684.                tabController.animateTo(0);
3685.                (context as
                Element).markNeedsBuild();
3686.              }
3687.            },
3688.          ),
3689.          FooterLink(
3690.            text: 'Appointments',
3691.            onTap: () {
3692.              final tabController =
                DefaultTabController.of(
3693.                  context,
3694.                );
3695.                if (tabController != null) {
3696.                  tabController.animateTo(1);
3697.                  (context as
                  Element).markNeedsBuild();
3698.                }
3699.              },
3700.            ),
3701.            FooterLink(
3702.              text: 'Feedback',
3703.              onTap: () {
3704.                final tabController =
                  DefaultTabController.of(
3705.                      context,
3706.                    );
3707.                    if (tabController != null) {
3708.                      tabController.animateTo(2);
3709.                      (context as
                      Element).markNeedsBuild();
3710.                    }
3711.                  },
3712.                ),
3713.                FooterLink(
3714.                  text: 'Contact Us',
3715.                  onTap: () {
3716.                    // TODO: Implement scroll
                    to contact section
3717.                    ScaffoldMessenger.of(context).showSnackBar(
3718.                      SnackBar(
3719.                        content: Text('Contact
                        Us clicked!'),
3720.                      ),
3721.                    );
3722.                  },
3723.                ),
3724.              ],
3725.            ),
3726.          ),
3727.          SizedBox(width: 24),
3728.          Expanded(
3729.            child: Column(
3730.              crossAxisAlignment:
                CrossAxisAlignment.start,
3731.              children: [
3732.                Text(
3733.                  'Contact',
3734.                  style: GoogleFonts.poppins(
3735.                    color: Colors.white,
3736.                    fontSize: 16,
3737.                    fontWeight:
                      FontWeight.w500,
3738.                  ),
3739.                ),
3740.                SizedBox(height: 12),
3741.                FooterLink(
3742.                  text: 'J. Yulo Avenue, Brgy.
                  Canlubang',
3743.                  icon: Icons.location_on,
3744.                ),
3745.                FooterLink(
3746.                  text: '(049) 520-5626',
3747.                  icon: Icons.phone,
3748.                  onTap: () async {
3749.                    final url =
                      Uri.parse('tel:(049) 520-5626');
3750.                    if (await
                      canLaunchUrl(url)) {
3751.                      await launchUrl(url);
3752.                    } else {
3753.                      ScaffoldMessenger.of(context).showSnackBar(
3754.                        SnackBar(
```

GCARE CONNECT main.dart

```
3755.         content: Text('Cannot
launch phone'),
3756.     ),
3757.     );
3758.   },
3759.   },
3760.   ),
3761.   FooterLink(
3762.     text:
'gcmccanlubang@gmail.com',
3763.     icon: Icons.email,
3764.     onTap: () async {
3765.       final url = Uri.parse(
3766.         'mailto:gcmccanlubang@gmail.com',
3767.       );
3768.       if (await
canLaunchUrl(url)) {
3769.         await launchUrl(url);
3770.       } else {
3771.         ScaffoldMessenger.of(context).showSnackBar(
3772.           SnackBar(
3773.             content: Text('Cannot
launch email'),
3774.           ),
3775.         );
3776.       }
3777.     },
3778.   ),
3779.   ],
3780. ),
3781. ),
3782. ],
3783. ),
3784.   SizedBox(height: 16),
3785.   Divider(color:
Colors.white.withOpacity(0.3)),
3786.   SizedBox(height: 16),
3787.   Text(
3788.     '© 2025 Global Care Medical
Center. All rights reserved.',
3789.     style: GoogleFonts.poppins(
3790.       color:
Colors.white.withOpacity(0.7),
3791.       fontSize: 14,
3792.     ),
3793.   ),
3794. ],
3795. ),
3796. );
3797. },
3798. );
3799. }
3800. }
3801.
3802. class FooterLink extends StatelessWidget {
3803.   final String text;
3804.   final IconData? icon;
3805.   final VoidCallback? onTap;
3806.
3807.   FooterLink({required this.text, this.icon,
this.onTap});
3808.
3809.   @override
3810.   Widget build(BuildContext context) {
3811.     return InkWell(
3812.       onTap: onTap,
3813.       child: Padding(
3814.         padding: EdgeInsets.symmetric(vertical: 6),
3815.         child: Row(
3816.           children: [
3817.             if (icon != null) ...[
3818.               Icon(icon, color:
Colors.white.withOpacity(0.9), size: 16),
3819.               SizedBox(width: 8),
3820.             ],
3821.             Text(
3822.               text,
3823.               style: GoogleFonts.poppins(
3824.                 color: Colors.white.withOpacity(0.9),
3825.                 fontSize: 14,
3826.                 decoration: onTap != null
? TextDecoration.underline
: TextDecoration.none,
3827.               ),
3828.             ),
3829.           ],
3830.         ),
3831.       ],
3832.     ),
3833.   ),
3834. );
3835. }
3836. }
3837.
3838. class SectionHeader extends StatelessWidget {
3839.   final String title;
3840.   final String? subtitle;
3841.
3842.   SectionHeader({required this.title, this.subtitle});
3843.
3844.   @override
3845.   Widget build(BuildContext context) {
3846.     return LayoutBuilder(
3847.       builder: (context, constraints) {
3848.         final isMobile = constraints.maxWidth < 600;
3849.         return Column(
3850.           children: [
3851.             AnimatedOpacity(
3852.               opacity: 1.0,
3853.               duration: Duration(milliseconds: 800),
3854.               child: Text(
3855.                 title,
3856.                 style: GoogleFonts.poppins(
3857.                   fontSize: isMobile ? 24 : 28,
3858.                   fontWeight: FontWeight.w600,
3859.                   color: Color(0xFF1F2937),
3860.                 ),
3861.               ),
3862.             ),
3863.             if (subtitle != null) ...[
3864.               SizedBox(height: isMobile ? 8 : 12),
3865.               AnimatedOpacity(
3866.                 opacity: 1.0,
3867.                 duration: Duration(milliseconds: 1000),
3868.                 child: Text(
3869.                   subtitle!,
3870.                   style: GoogleFonts.poppins(
3871.                     fontSize: isMobile ? 14 : 16,
3872.                     fontWeight: FontWeight.w400,
3873.                     color: Color(0xFF6B7280),
3874.                   ),
3875.                 ),
3876.               ],
3877.             ],
3878.           ],
3879.         ),
3880.       ),
3881.     );
3882.   }
3883. }
```

GCARE CONNECT main.dart

```
3876.         ),
3877.       ),
3878.     ],
3879.   ],
3880. );
3881. },
3882. );
3883. }
3884. }
3885.
3886. class DashboardSection extends StatelessWidget {
3887.   @override
3888.   Widget build(BuildContext context) {
3889.     return Consumer<AuthService>(
3890.       builder: (context, authService, child) {
3891.         return LayoutBuilder(
3892.           builder: (context, constraints) {
3893.             final double screenWidth =
constraints.maxWidth;
3894.             final bool isMobile = screenWidth < 600;
3895.             final bool isTablet = screenWidth >= 600
&& screenWidth < 900;
3896.             final double cardPadding = isMobile
? 12.0
3897.             : isTablet
? 16.0
3898.             : 20.0;
3899.             final double sectionSpacing = isMobile
? 12.0
3900.             : isTablet
? 16.0
3901.             : 20.0;
3902.             final double chartHeight = isMobile
? 180.0
3903.             : isTablet
? 220.0
3904.             : 260.0;
3905.             final double fontSizeTitle = isMobile
? 16.0
3906.             : isTablet
? 18.0
3907.             : 20.0;
3908.             final double fontSizeSubtitle = isMobile
? 12.0
3909.             : isTablet
? 14.0
3910.             : 16.0;
3911.             return Container(
3912.               color: const Color(0xFFFF8FAFC),
3913.               padding: EdgeInsets.symmetric(
3914.                 vertical: isMobile
? 16.0
3915.                 : isTablet
? 24.0
3916.                 : 32.0,
3917.                 horizontal: screenWidth * 0.05,
3918.               ),
3919.               child: SingleChildScrollView(
3920.                 child: Column(
3921.                   crossAxisAlignment:
CrossAxisAlignment.center,
3922.                   children: [
3923.                     // Welcome Banner
3924.                     Center(
3925.                       child: Container(
3939.                         constraints:
BoxConstraints(maxWidth: 600),
3940.                         child: _buildWelcomeBanner(
3941.                           context,
3942.                           authService,
3943.                           isMobile,
3944.                           fontSizeTitle,
3945.                           fontSizeSubtitle,
3946.                         ),
3947.                       ),
3948.                     ),
3949.                     SizedBox(height: sectionSpacing),
3950.                     // If not logged in
3951.                     if (!authService.isLoggedIn)
3952.                     Center(
3953.                       child: Container(
3954.                         constraints:
BoxConstraints(maxWidth: 600),
3955.                         child: _buildLoginPrompt(
3956.                           context,
3957.                           isMobile,
3958.                           cardPadding,
3959.                           fontSizeTitle,
3960.                         ),
3961.                       ),
3962.                     ),
3963.                     else
3964.                     Column(
3965.                       crossAxisAlignment:
CrossAxisAlignment.center,
3966.                       children: [
3967.                         // Overview Section
3968.                         Center(
3969.                           child: SectionHeader(
3970.                             title: 'Dashboard Overview',
3971.                             subtitle:
'Your healthcare insights
and management hub.',
3972.                           ),
3973.                         ),
3974.                       ],
3975.                     ),
3976.                     SizedBox(height:
sectionSpacing),
3977.                     _buildOverviewCharts(
3978.                       context,
3979.                       isMobile,
3980.                       isTablet,
3981.                       cardPadding,
3982.                       chartHeight,
3983.                       fontSizeSubtitle,
3984.                       screenWidth,
3985.                     ),
3986.                     SizedBox(height:
sectionSpacing),
3987.                     // User Profile
3988.                     _buildUserProfile(
3989.                       context,
3990.                       authService,
3991.                       isMobile,
3992.                       cardPadding,
3993.                       fontSizeTitle,
3994.                       fontSizeSubtitle,
3995.                     ),
3996.                   ],
3997.                 ),
3998.               ),
3999.             ),
4000.           ),
4001.         ),
4002.       ),
4003.     ),
4004.   ),
4005. );
```

GCARE CONNECT main.dart

```
3998.      SizedBox(height:
sectionSpacing),
3999.      // Appointments
4000.      Text(
4001.        'Your Appointments',
4002.        style: GoogleFonts.poppins(
4003.          fontSize: fontSizeTitle,
4004.          fontWeight:
FontWeight.w600,
4005.          color: const
Color(0xFF1F2937),
4006.        ),
4007.      ),
4008.      SizedBox(height: sectionSpacing
/ 2),
4009.      AppointmentHistory(isMobile:
isMobile),
4010.
4011.      SizedBox(height:
sectionSpacing),
4012.      // Feedback
4013.      Text(
4014.        'Your Feedback',
4015.        style: GoogleFonts.poppins(
4016.          fontSize: fontSizeTitle,
4017.          fontWeight:
FontWeight.w600,
4018.          color: const
Color(0xFF1F2937),
4019.        ),
4020.      ),
4021.      SizedBox(height: sectionSpacing
/ 2),
4022.      _buildFeedbackSection(
4023.        context,
4024.        isMobile,
4025.        cardPadding,
4026.        fontSizeSubtitle,
4027.      ),
4028.    ],
4029.  ),
4030. ],
4031. ),
4032. ),
4033. );
4034. },
4035. );
4036. },
4037. );
4038. }
4039.
4040. Widget _buildWelcomeBanner(
4041.   BuildContext context,
4042.   AuthService authService,
4043.   bool isMobile,
4044.   double fontSizeTitle,
4045.   double fontSizeSubtitle,
4046. ) {
4047.   return Card(
4048.     elevation: 6,
4049.     shape:
RoundedRectangleBorder(borderRadius:
BorderRadius.circular(12)),
4050.     child: Container(
4051.       decoration: BoxDecoration(
4052.         gradient: LinearGradient(
4053.           colors: [Colors.indigo[700]!,
Colors.indigo[500]!],
4054.           begin: Alignment.topLeft,
4055.           end: Alignment.bottomRight,
4056.         ),
4057.         borderRadius: BorderRadius.circular(12),
4058.       ),
4059.       padding: EdgeInsets.all(isMobile ? 12.0 :
16.0),
4060.       child: Column(
4061.         mainAxisAlignment: MainAxisAlignment.min,
4062.         crossAxisAlignment:
CrossAxisAlignment.center,
4063.         children: [
4064.           Row(
4065.             mainAxisAlignment:
MainAxisAlignment.center,
4066.             children: [
4067.               Text(
4068.                 'Welcome${authService.isLoggedIn ?
' ', ${authService.username}' : ''}!',
4069.                 style: GoogleFonts.poppins(
4070.                   color: Colors.white,
4071.                   fontSize: fontSizeTitle,
4072.                   fontWeight: FontWeight.w600,
4073.                 ),
4074.                 textAlign: TextAlign.center,
4075.               ),
4076.               SizedBox(width: isMobile ? 8.0 : 12.0),
4077.               Icon(
4078.                 Icons.favorite,
4079.                 color: Colors.white,
4080.                 size: isMobile ? 28.0 : 32.0,
4081.               ),
4082.             ],
4083.           ),
4084.           SizedBox(height: 8),
4085.           Text(
4086.             'Your health, our priority. Manage your
care with ease.',
4087.             style: GoogleFonts.poppins(
4088.               color: Colors.white.withOpacity(0.9),
4089.               fontSize: fontSizeSubtitle,
4090.             ),
4091.             textAlign: TextAlign.center,
4092.           ),
4093.         ],
4094.       ),
4095.     ),
4096.   );
4097. }
4098.
4099. Widget _buildLoginPrompt(
4100.   BuildContext context,
4101.   bool isMobile,
4102.   double cardPadding,
4103.   double fontSizeTitle,
4104. ) {
4105.   return Card(
4106.     elevation: 4,
4107.     shape:
RoundedRectangleBorder(borderRadius:
BorderRadius.circular(12)),
4108.     child: Padding(
4109.       padding: EdgeInsets.all(cardPadding),
4110.       child: Column(
```

GCARE CONNECT main.dart

```

4111.      mainAxisAlignment:
MainAxisAlignment.center,
4112.      children: [
4113.        Icon(
4114.          Icons.lock,
4115.          size: isMobile ? 28.0 : 32.0,
4116.          color: Colors.indigo[700],
4117.        ),
4118.        SizedBox(height: 12),
4119.        Text(
4120.          'Please login or sign up to view your
dashboard.',
4121.          style: GoogleFonts.poppins(
4122.            fontSize: fontSizeTitle,
4123.            fontWeight: FontWeight.w500,
4124.            color: const Color(0xFF1F2937),
4125.          ),
4126.          textAlign: TextAlign.center,
4127.        ),
4128.        SizedBox(height: 16),
4129.        ElevatedButton(
4130.          onPressed: () {
4131.            showDialog(
4132.              context: context,
4133.              builder: (context) => const
AuthDialog(),
4134.            );
4135.          },
4136.          child: Text('Login / Sign Up'),
4137.          style: ElevatedButton.styleFrom(
4138.            padding: EdgeInsets.symmetric(
4139.              horizontal: isMobile ? 20.0 : 24.0,
4140.              vertical: isMobile ? 10.0 : 12.0,
4141.            ),
4142.          ),
4143.        ),
4144.      ],
4145.    ),
4146.  ),
4147. );
4148. }
4149.
4150. Widget _buildUserProfile(
4151.   BuildContext context,
4152.   AuthService authService,
4153.   bool isMobile,
4154.   double cardPadding,
4155.   double fontSizeTitle,
4156.   double fontSizeSubtitle,
4157. ) {
4158.   return Card(
4159.     elevation: 4,
4160.     shape:
RoundedRectangleBorder(borderRadius:
BorderRadius.circular(12)),
4161.     child: Padding(
4162.       padding: EdgeInsets.all(cardPadding),
4163.       child: Row(
4164.         children: [
4165.           CircleAvatar(
4166.             backgroundColor:
Colors.indigo.withOpacity(0.1),
4167.             child: Icon(
4168.               Icons.person,
4169.               color: Colors.indigo[700],
4170.               size: isMobile ? 24.0 : 28.0,
4171.             ),
4172.             radius: isMobile ? 24.0 : 28.0,
4173.           ),
4174.           SizedBox(width: 12.0),
4175.           Expanded(
4176.             child: Column(
4177.               crossAxisAlignment:
CrossAxisAlignment.start,
4178.               children: [
4179.                 Text(
4180.                   authService.username ?? 'User',
4181.                   style: GoogleFonts.poppins(
4182.                     fontSize: fontSizeTitle,
4183.                     fontWeight: FontWeight.w600,
4184.                     color: const Color(0xFF1F2937),
4185.                   ),
4186.                 ),
4187.                 Text(
4188.                   authService.email ?? 'N/A',
4189.                   style: GoogleFonts.poppins(
4190.                     fontSize: fontSizeSubtitle,
4191.                     color: Colors.grey[600],
4192.                   ),
4193.                 ),
4194.               ],
4195.             ),
4196.           ),
4197.           OutlinedButton.icon(
4198.             onPressed: () {
4199.               authService.logout();
4200.               final tabController =
DefaultTabController.of(context);
4201.               if (tabController != null) {
4202.                 tabController.animateTo(0);
4203.                 (context as
Element).markNeedsBuild();
4204.               }
4205.             },
4206.             icon: Icon(Icons.logout, size: isMobile ?
16.0 : 18.0),
4207.             label: Text('Logout'),
4208.             style: OutlinedButton.styleFrom(
4209.               padding: EdgeInsets.symmetric(
4210.                 horizontal: isMobile ? 12.0 : 16.0,
4211.                 vertical: isMobile ? 6.0 : 8.0,
4212.               ),
4213.             ),
4214.           ),
4215.         ],
4216.       ),
4217.     ),
4218.   );
4219. }
4220.
4221. Widget _buildOverviewCharts(
4222.   BuildContext context,
4223.   bool isMobile,
4224.   bool isTablet,
4225.   double cardPadding,
4226.   double chartHeight,
4227.   double fontSizeSubtitle,
4228.   double screenWidth,
4229. ) {
4230.   final int crossAxisCount = isMobile
? 1
4231.   : isTablet
4232.

```

GCARE CONNECT main.dart

```
4233.      ? 2
4234.      : 3;
4235.      return Card(
4236.        elevation: 4,
4237.        shape:
          RoundedRectangleBorder(borderRadius:
            BorderRadius.circular(12)),
4238.        child: Padding(
4239.          padding: EdgeInsets.all(cardPadding),
4240.          child: GridView.count(
4241.            shrinkWrap: true,
4242.            physics: const
              NeverScrollableScrollPhysics(),
4243.            crossAxisCount: crossAxisCount,
4244.            crossAxisSpacing: isMobile ? 12.0 : 16.0,
4245.            mainAxisSpacing: isMobile ? 12.0 : 16.0,
4246.            childAspectRatio: isMobile
              ? 1.2
4247.              : isTablet
4248.              ? 1.1
4249.              : 1.0,
4250.            children: _buildChartWidgets(
4251.              context,
4252.              isMobile,
4253.              chartHeight,
4254.              fontSizeSubtitle,
4255.            ),
4256.          ),
4257.        ),
4258.      );
4259.    };
4260.  }
4261.
4262.  List<Widget> _buildChartWidgets(
4263.    BuildContext context,
4264.    bool isMobile,
4265.    double chartHeight,
4266.    double fontSizeSubtitle,
4267.  ) {
4268.    return [
4269.      _buildChartCard(
4270.        context: context,
4271.        isMobile: isMobile,
4272.        title: 'Appointments by Department',
4273.        chart: _buildDepartmentChart(isMobile,
          chartHeight),
4274.        fontSizeSubtitle: fontSizeSubtitle,
4275.      ),
4276.      _buildChartCard(
4277.        context: context,
4278.        isMobile: isMobile,
4279.        title: 'Patient Satisfaction',
4280.        chart: _buildSatisfactionChart(isMobile,
          chartHeight),
4281.        fontSizeSubtitle: fontSizeSubtitle,
4282.      ),
4283.      _buildChartCard(
4284.        context: context,
4285.        isMobile: isMobile,
4286.        title: 'Appointment Status',
4287.        chart: _buildStatusChart(isMobile,
          chartHeight),
4288.        fontSizeSubtitle: fontSizeSubtitle,
4289.      ),
4290.    ];
4291.  }
4292.
4293.  Widget _buildChartCard({
4294.    required BuildContext context,
4295.    required bool isMobile,
4296.    required String title,
4297.    required Widget chart,
4298.    required double fontSizeSubtitle,
4299.  }) {
4300.    return Column(
4301.      crossAxisAlignment: CrossAxisAlignment.start,
4302.      children: [
4303.        Text(
4304.          title,
4305.          style: GoogleFonts.poppins(
4306.            fontSize: fontSizeSubtitle,
4307.            fontWeight: FontWeight.w600,
4308.            color: const Color(0xFF1F2937),
4309.          ),
4310.        ),
4311.        SizedBox(height: 8),
4312.        Container(
4313.          height: isMobile ? 180.0 : 220.0,
4314.          decoration: BoxDecoration(
4315.            color: Colors.white,
4316.            borderRadius: BorderRadius.circular(12),
4317.            boxShadow: [
4318.              BoxShadow(
4319.                color: Colors.black.withOpacity(0.05),
4320.                blurRadius: 8,
4321.                offset: const Offset(0, 2),
4322.              ),
4323.            ],
4324.          ),
4325.          padding: const EdgeInsets.all(12),
4326.          child: chart,
4327.        ),
4328.      ],
4329.    );
4330.  }
4331.
4332.  Widget _buildDepartmentChart(bool isMobile,
    double chartHeight) {
4333.    return SizedBox(
4334.      height: chartHeight,
4335.      child: BarChart(
4336.        BarChartData(
4337.          alignment:
            BarChartAlignment.spaceAround,
4338.          maxY: 200,
4339.          barTouchData: BarTouchData(
4340.            enabled: true,
4341.            touchTooltipData: BarTouchTooltipData(
4342.              tooltipMargin: 10,
4343.              getTooltipItem: (group, groupIndex,
                rod, rodIndex) {
4344.                const titles = ['Cardio', 'Neuro', 'Ortho',
                  'Pedia', 'Derma'];
4345.                return BarTooltipItem(
4346.                  '${titles[group.x]}\n${rod.toY.toInt()}',
4347.                  GoogleFonts.poppins(
4348.                    color: Colors.white,
4349.                    fontSize: isMobile ? 10 : 12,
4350.                    fontWeight: FontWeight.w500,
4351.                  ),
4352.                  textAlign: TextAlign.center,
4353.                  children: [],
```

GCARE CONNECT main.dart

```
4354.         );
4355.       },
4356.       getTooltipColor: (group) =>
Colors.indigo[700]!.withOpacity(0.8),
4357.     ),
4358.   ),
4359.   titlesData: FITitlesData(
4360.     show: true,
4361.     bottomTitles: AxisTitles(
4362.       sideTitles: SideTitles(
4363.         showTitles: true,
4364.         getTitlesWidget: (value, meta) {
4365.           const titles = ['Cardio', 'Neuro',
'Ortho', 'Pedia', 'Derma'];
4366.           return Text(
4367.             titles[value.toInt()],
4368.             style: GoogleFonts.poppins(
4369.               fontSize: isMobile ? 8 : 10,
4370.               color: Colors.grey[600],
4371.             ),
4372.           );
4373.         },
4374.       ),
4375.     ),
4376.     leftTitles: AxisTitles(
4377.       sideTitles: SideTitles(
4378.         showTitles: true,
4379.         interval: 50,
4380.         getTitlesWidget: (value, meta) {
4381.           return Text(
4382.             value.toInt().toString(),
4383.             style: GoogleFonts.poppins(
4384.               fontSize: isMobile ? 8 : 10,
4385.               color: Colors.grey[600],
4386.             ),
4387.           );
4388.         },
4389.       ),
4390.     ),
4391.     topTitles: const AxisTitles(
4392.       sideTitles: SideTitles(showTitles: false),
4393.     ),
4394.     rightTitles: const AxisTitles(
4395.       sideTitles: SideTitles(showTitles: false),
4396.     ),
4397.   ),
4398.   borderData: FIBorderData(show: false),
4399.   barGroups: [
4400.     BarChartGroupData(
4401.       x: 0,
4402.       barRods: [
4403.         BarChartRodData(
4404.           toY: 120,
4405.           color: Colors.indigo[700],
4406.           width: isMobile ? 8 : 12,
4407.           borderRadius:
BorderRadius.circular(4),
4408.         ),
4409.       ],
4410.     ),
4411.     BarChartGroupData(
4412.       x: 1,
4413.       barRods: [
4414.         BarChartRodData(
4415.           toY: 80,
4416.           color: Colors.indigo[600],
4417.           width: isMobile ? 8 : 12,
4418.           borderRadius:
BorderRadius.circular(4),
4419.         ),
4420.       ],
4421.     ),
4422.     BarChartGroupData(
4423.       x: 2,
4424.       barRods: [
4425.         BarChartRodData(
4426.           toY: 100,
4427.           color: Colors.indigo[500],
4428.           width: isMobile ? 8 : 12,
4429.           borderRadius:
BorderRadius.circular(4),
4430.         ),
4431.       ],
4432.     ),
4433.     BarChartGroupData(
4434.       x: 3,
4435.       barRods: [
4436.         BarChartRodData(
4437.           toY: 150,
4438.           color: Colors.indigo[400],
4439.           width: isMobile ? 8 : 12,
4440.           borderRadius:
BorderRadius.circular(4),
4441.         ),
4442.       ],
4443.     ),
4444.     BarChartGroupData(
4445.       x: 4,
4446.       barRods: [
4447.         BarChartRodData(
4448.           toY: 60,
4449.           color: Colors.indigo[300],
4450.           width: isMobile ? 8 : 12,
4451.           borderRadius:
BorderRadius.circular(4),
4452.         ),
4453.       ],
4454.     ),
4455.   ],
4456. ),
4457. ),
4458. );
4459. }
4460.
4461. Widget _buildSatisfactionChart(bool isMobile,
double chartHeight) {
4462.   return SizedBox(
4463.     height: chartHeight,
4464.     child: BarChart(
4465.       BarChartData(
4466.         alignment:
BarChartAlignment.spaceAround,
4467.         maxY: 5,
4468.         barTouchData: BarTouchData(
4469.           enabled: true,
4470.           touchTooltipData: BarTouchTooltipData(
4471.             tooltipMargin: 10,
4472.             getTooltipItem: (group, groupIndex,
rod, rodIndex) {
4473.               const titles = ['Jan', 'Feb', 'Mar', 'Apr',
'May', 'Jun'];
4474.               return BarTooltipItem(
```

GCARE CONNECT main.dart

```
4475.      '${titles[group.x]}\n${rod.toY.toStringAsFixed(1)}',
4476.      GoogleFonts.poppins(
4477.        color: Colors.white,
4478.        fontSize: isMobile ? 10 : 12,
4479.        fontWeight: FontWeight.w500,
4480.      ),
4481.      textAlign: TextAlign.center,
4482.      children: [],
4483.    );
4484.  },
4485.  getTooltipColor: (group) =>
    Colors.indigo[700]!.withOpacity(0.8),
4486.  ),
4487.  ),
4488.  titlesData: FITitlesData(
4489.    show: true,
4490.    bottomTitles: AxisTitles(
4491.      sideTitles: SideTitles(
4492.        showTitles: true,
4493.        getTitlesWidget: (value, meta) {
4494.          const titles = ['Jan', 'Feb', 'Mar', 'Apr',
            'May', 'Jun'];
4495.          return Text(
4496.            titles[value.toInt()],
4497.            style: GoogleFonts.poppins(
4498.              fontSize: isMobile ? 8 : 10,
4499.              color: Colors.grey[600],
4500.            ),
4501.          );
4502.        },
4503.      ),
4504.    ),
4505.    leftTitles: AxisTitles(
4506.      sideTitles: SideTitles(
4507.        showTitles: true,
4508.        interval: 1,
4509.        getTitlesWidget: (value, meta) {
4510.          return Text(
4511.            value.toInt().toString(),
4512.            style: GoogleFonts.poppins(
4513.              fontSize: isMobile ? 8 : 10,
4514.              color: Colors.grey[600],
4515.            ),
4516.          );
4517.        },
4518.      ),
4519.    ),
4520.    topTitles: const AxisTitles(
4521.      sideTitles: SideTitles(showTitles: false),
4522.    ),
4523.    rightTitles: const AxisTitles(
4524.      sideTitles: SideTitles(showTitles: false),
4525.    ),
4526.  ),
4527.  borderData: FIBorderData(show: false),
4528.  barGroups: [
4529.    BarChartGroupData(
4530.      x: 0,
4531.      barRods: [
4532.        BarChartRodData(
4533.          toY: 4.2,
4534.          color: Colors.indigo[700],
4535.          width: isMobile ? 8 : 12,
4536.          borderRadius:
            BorderRadius.circular(4),
4537.        ),
4538.      ],
4539.    ),
4540.    BarChartGroupData(
4541.      x: 1,
4542.      barRods: [
4543.        BarChartRodData(
4544.          toY: 4.3,
4545.          color: Colors.indigo[600],
4546.          width: isMobile ? 8 : 12,
4547.          borderRadius:
            BorderRadius.circular(4),
4548.        ),
4549.      ],
4550.    ),
4551.    BarChartGroupData(
4552.      x: 2,
4553.      barRods: [
4554.        BarChartRodData(
4555.          toY: 4.5,
4556.          color: Colors.indigo[500],
4557.          width: isMobile ? 8 : 12,
4558.          borderRadius:
            BorderRadius.circular(4),
4559.        ),
4560.      ],
4561.    ),
4562.    BarChartGroupData(
4563.      x: 3,
4564.      barRods: [
4565.        BarChartRodData(
4566.          toY: 4.1,
4567.          color: Colors.indigo[400],
4568.          width: isMobile ? 8 : 12,
4569.          borderRadius:
            BorderRadius.circular(4),
4570.        ),
4571.      ],
4572.    ),
4573.    BarChartGroupData(
4574.      x: 4,
4575.      barRods: [
4576.        BarChartRodData(
4577.          toY: 4.4,
4578.          color: Colors.indigo[300],
4579.          width: isMobile ? 8 : 12,
4580.          borderRadius:
            BorderRadius.circular(4),
4581.        ),
4582.      ],
4583.    ),
4584.    BarChartGroupData(
4585.      x: 5,
4586.      barRods: [
4587.        BarChartRodData(
4588.          toY: 4.6,
4589.          color: Colors.indigo[200],
4590.          width: isMobile ? 8 : 12,
4591.          borderRadius:
            BorderRadius.circular(4),
4592.        ),
4593.      ],
4594.    ),
4595.  ],
4596.  ),
4597.  ),
```


GCARE CONNECT main.dart

```
4598.         );
4599.       }
4600.
4601.       Widget _buildStatusChart(bool isMobile, double
chartHeight) {
4602.         return SizedBox(
4603.           height: chartHeight,
4604.           child: BarChart(
4605.             BarChartData(
4606.               alignment:
BarChartAlignment.spaceAround,
4607.               maxY: 250,
4608.               barTouchData: BarTouchData(
4609.                 enabled: true,
4610.                 touchTooltipData: BarTouchTooltipData(
4611.                   tooltipMargin: 10,
4612.                   getTooltipItem: (group, groupIndex,
rod, rodIndex) {
4613.                     const titles = ['Upcoming', 'Completed',
'Cancelled'];
4614.                     return BarTooltipItem(
4615.                       '${titles[group.x]}\n${rod.toY.toInt()}',
4616.                       GoogleFonts.poppins(
4617.                         color: Colors.white,
4618.                         fontSize: isMobile ? 10 : 12,
4619.                         fontWeight: FontWeight.w500,
4620.                       ),
4621.                       textAlign: TextAlign.center,
4622.                       children: [],
4623.                     );
4624.                   },
4625.                   getTooltipColor: (group) =>
Colors.indigo[700]!.withOpacity(0.8),
4626.                 ),
4627.               ),
4628.               titlesData: FITitlesData(
4629.                 show: true,
4630.                 bottomTitles: AxisTitles(
4631.                   sideTitles: SideTitles(
4632.                     showTitles: true,
4633.                     getTitlesWidget: (value, meta) {
4634.                       const titles = ['Upcoming',
'Completed', 'Cancelled'];
4635.                       return Text(
4636.                         titles[value.toInt()],
4637.                         style: GoogleFonts.poppins(
4638.                           fontSize: isMobile ? 8 : 10,
4639.                           color: Colors.grey[600],
4640.                         ),
4641.                       );
4642.                     },
4643.                   ),
4644.                 ),
4645.                 leftTitles: AxisTitles(
4646.                   sideTitles: SideTitles(
4647.                     showTitles: true,
4648.                     interval: 50,
4649.                     getTitlesWidget: (value, meta) {
4650.                       return Text(
4651.                         value.toInt().toString(),
4652.                         style: GoogleFonts.poppins(
4653.                           fontSize: isMobile ? 8 : 10,
4654.                           color: Colors.grey[600],
4655.                         ),
4656.                       );
4657.                     },
4658.                   ),
4659.                 ),
4660.                 topTitles: const AxisTitles(
4661.                   sideTitles: SideTitles(showTitles: false),
4662.                 ),
4663.                 rightTitles: const AxisTitles(
4664.                   sideTitles: SideTitles(showTitles: false),
4665.                 ),
4666.               ),
4667.               borderData: FIBorderData(show: false),
4668.               barGroups: [
4669.                 BarChartGroupData(
4670.                   x: 0,
4671.                   barRods: [
4672.                     BarChartRodData(
4673.                       toY: 50,
4674.                       color: Colors.indigo[700],
4675.                       width: isMobile ? 8 : 12,
4676.                       borderRadius:
BorderRadius.circular(4),
4677.                     ),
4678.                   ],
4679.                 ),
4680.                 BarChartGroupData(
4681.                   x: 1,
4682.                   barRods: [
4683.                     BarChartRodData(
4684.                       toY: 200,
4685.                       color: Colors.indigo[600],
4686.                       width: isMobile ? 8 : 12,
4687.                       borderRadius:
BorderRadius.circular(4),
4688.                     ),
4689.                   ],
4690.                 ),
4691.                 BarChartGroupData(
4692.                   x: 2,
4693.                   barRods: [
4694.                     BarChartRodData(
4695.                       toY: 30,
4696.                       color: Colors.indigo[500],
4697.                       width: isMobile ? 8 : 12,
4698.                       borderRadius:
BorderRadius.circular(4),
4699.                     ),
4700.                   ],
4701.                 ),
4702.               ],
4703.             ),
4704.           );
4705.         };
4706.       }
4707.
4708.       Widget _buildFeedbackSection(
4709.         BuildContext context,
4710.         bool isMobile,
4711.         double cardPadding,
4712.         double fontSizeSubtitle,
4713.       ) {
4714.         return FutureBuilder<List<Map<String,
dynamic>>>>(>
4715.           future:
Provider.of<FeedbackService>(context).loadFeedback(),
4716.           builder: (context, snapshot) {
```

GCARE CONNECT main.dart

```
4717.         if (snapshot.connectionState ==
ConnectionState.waiting) {
4718.             return const Center(child:
CircularProgressIndicator());
4719.         }
4720.         if (snapshot.hasError) {
4721.             return Card(
4722.                 elevation: 4,
4723.                 shape: RoundedRectangleBorder(
4724.                     borderRadius: BorderRadius.circular(12),
4725.                 ),
4726.                 child: Padding(
4727.                     padding: EdgeInsets.all(cardPadding),
4728.                     child: Center(
4729.                         child: Text(
4730.                             'Error loading feedback',
4731.                             style: GoogleFonts.poppins(
4732.                                 fontSize: fontSizeSubtitle,
4733.                                 color: const Color(0xFF1F2937),
4734.                             ),
4735.                         ),
4736.                     ),
4737.                 );
4738.             };
4739.         }
4740.         final feedbackList = snapshot.data ?? [];
4741.         if (feedbackList.isEmpty) {
4742.             return Card(
4743.                 elevation: 4,
4744.                 shape: RoundedRectangleBorder(
4745.                     borderRadius: BorderRadius.circular(12),
4746.                 ),
4747.                 child: Padding(
4748.                     padding: EdgeInsets.all(cardPadding),
4749.                     child: Center(
4750.                         child: Text(
4751.                             'No feedback submitted yet.',
4752.                             style: GoogleFonts.poppins(
4753.                                 fontSize: fontSizeSubtitle,
4754.                                 color: const Color(0xFF1F2937),
4755.                             ),
4756.                         ),
4757.                     ),
4758.                 );
4759.             };
4760.         }
4761.         return ListView.separated(
4762.             shrinkWrap: true,
4763.             physics: const
NeverScrollableScrollPhysics(),
4764.             itemCount: feedbackList.length,
4765.             separatorBuilder: (context, index) => const
 SizedBox(height: 12),
4766.             itemBuilder: (context, index) {
4767.                 final feedback = feedbackList[index];
4768.                 return Card(
4769.                     elevation: 4,
4770.                     shape: RoundedRectangleBorder(
4771.                         borderRadius:
BorderRadius.circular(12),
4772.                     ),
4773.                     child: Padding(
4774.                         padding: EdgeInsets.all(isMobile ? 10.0
: 12.0),
4775.                     child: Column(
4776.                         crossAxisAlignment:
CrossAxisAlignment.start,
4777.                         children: [
4778.                             Row(
4779.                                 children: [
4780.                                     Icon(
4781.                                         Icons.star,
4782.                                         color: Colors.amber,
4783.                                         size: isMobile ? 18.0 : 20.0,
4784.                                     ),
4785.                                     const SizedBox(width: 8),
4786.                                     Text(
4787.                                         'Rating:
${feedback['rating']}.toStringAsFixed(1)}',
4788.                                         style: GoogleFonts.poppins(
4789.                                             fontSize: fontSizeSubtitle,
4790.                                             fontWeight: FontWeight.w600,
4791.                                             color: const
Color(0xFF1F2937),
4792.                                         ),
4793.                                     ),
4794.                                 ],
4795.                             ),
4796.                             const SizedBox(height: 8),
4797.                             Text(
4798.                                 feedback['comment'] ?? 'N/A',
4799.                                 style: GoogleFonts.poppins(
4800.                                     fontSize: fontSizeSubtitle,
4801.                                     color: Colors.grey[600],
4802.                                 ),
4803.                             ),
4804.                             const SizedBox(height: 8),
4805.                             Text(
4806.                                 'Submitted on: ${feedback['date']
?? 'N/A'}',
4807.                                 style: GoogleFonts.poppins(
4808.                                     fontSize: isMobile ? 10.0 : 12.0,
4809.                                     color: Colors.grey[600],
4810.                                 ),
4811.                             ),
4812.                         ],
4813.                     ),
4814.                 );
4815.             };
4816.         },
4817.     );
4818.     },
4819. );
4820. }
4821. }
```