**📄 1\_home\_page.dart (DART, 266 lines)**

1 │ import 'package:dkkb\_app/4\_presention/Pages/drawer/custom\_drawer.dart';

2 │ import 'package:dkkb\_app/4\_presention/Pages/home/screen/3c\_tailieusongkhoe\_section.dart';

3 │ import 'package:dkkb\_app/4\_presention/Pages/home/screen/3d\_ads\_list\_section.dart';

4 │ import 'package:dkkb\_app/4\_presention/Pages/home/screen/3b\_doctors\_section.dart';

5 │ import 'package:dkkb\_app/4\_presention/Pages/home/widget/home\_icons\_section.dart';

6 │ import 'package:dkkb\_app/4\_presention/Pages/home/screen/3a\_news\_list\_section.dart';

7 │ import 'package:dkkb\_app/5\_app/constant/showcase\_keys.dart';

8 │ import 'package:dkkb\_app/5\_app/constant/showcase\_text.dart';

9 │ import 'package:dkkb\_app/5\_app/widgets/appbar\_custom.dart';

10 │ import 'package:dkkb\_app/8\_service/animation/seasonal\_effect\_wrapper.dart';

11 │ import 'package:dkkb\_app/7\_core/responsive/app\_dimensions.dart';

12 │ //import 'package:dkkb\_app/8\_service/notification/android\_notification\_details.dart';

13 │ import 'package:dkkb\_app/commons.dart';

14 │ import 'package:flutter/material.dart';

15 │ import 'package:dkkb\_app/4\_presention/Pages/home/screen/2\_home\_listener.dart';

16 │ import 'package:dkkb\_app/4\_presention/bloc/home/home\_bloc.dart';

17 │ import 'package:dkkb\_app/5\_app/color/app\_colors.dart';

18 │ //import 'package:flutter\_local\_notifications/flutter\_local\_notifications.dart';

19 │ import 'package:get\_it/get\_it.dart';

20 │ import 'package:showcaseview/showcaseview.dart';

21 │

22 │ class HomePage extends StatelessWidget {

23 │ const HomePage({super.key});

24 │

25 │ @override

26 │ Widget build(BuildContext context) {

27 │ return ShowCaseWidget(

28 │ enableAutoScroll: true,

29 │ builder: (context) => BlocProvider(

30 │ create: (context) =>

31 │ GetIt.I<HomeBloc>()..add(const HomeEvent.started()),

32 │ child: const HomePageContent(),

33 │ ),

34 │ globalTooltipActionConfig: const TooltipActionConfig(

35 │ position: TooltipActionPosition.outside,

36 │ alignment: MainAxisAlignment.spaceBetween,

37 │ actionGap: 10,

38 │ ),

39 │ );

40 │ }

41 │ }

42 │

43 │ class HomePageContent extends StatefulWidget {

44 │ const HomePageContent({super.key});

45 │

46 │ @override

47 │ State<HomePageContent> createState() => \_HomePageContentState();

48 │ }

49 │

50 │ class \_HomePageContentState extends State<HomePageContent> {

51 │ final bool \_enableShowcase = false;

52 │ Future<void> \_handleRefresh() async {

53 │ try {

54 │ context.read<HomeBloc>().add(const HomeEvent.started(forceRefresh: true));

55 │ await Future.delayed(const Duration(seconds: 2));

56 │ } catch (e) {

57 │ debugPrint('Error refreshing home page: $e');

58 │ }

59 │ }

60 │

61 │ Widget \_buildPlaceholder(GlobalKey key) {

62 │ return Container(

63 │ key: key, // Sử dụng key riêng cho placeholder

64 │ width: 0,

65 │ height: 0,

66 │ // ignore: deprecated\_member\_use

67 │ color: Colors.black12.withValues(alpha: 0.5),

68 │ );

69 │ }

70 │

71 │ Widget \_buildShowcase({

72 │ required GlobalKey key,

73 │ required String title,

74 │ required String description,

75 │ required Widget child,

76 │ EdgeInsets? targetPadding, // Thêm tham số nullable

77 │ ShapeBorder? targetShapeBorder, // Thêm tham số nullable

78 │ bool? enableAutoScroll,

79 │ }) {

80 │ return Showcase(

81 │ key: key,

82 │ title: title,

83 │ description: description,

84 │ enableAutoScroll: enableAutoScroll ?? true,

85 │

86 │ // Tùy chỉnh overlay

87 │ overlayColor: Colors.black12,

88 │ overlayOpacity: 0.5,

89 │

90 │ // Sử dụng targetPadding từ tham số hoặc giá trị mặc định

91 │ targetPadding: targetPadding ?? const EdgeInsets.all(10),

92 │ // Sử dụng targetShapeBorder từ tham số hoặc giá trị mặc định

93 │ targetShapeBorder: targetShapeBorder ?? const RoundedRectangleBorder(),

94 │

95 │ // Tùy chỉnh tooltip

96 │

97 │ tooltipBorderRadius: BorderRadius.circular(10),

98 │ tooltipPadding: const EdgeInsets.all(15),

99 │

100 │ // Tùy chỉnh text

101 │ titleTextStyle:

102 │ const TextStyle(fontSize: 20, color: AppColors.textColorBlue),

103 │ descTextStyle: const TextStyle(fontSize: 16),

104 │

105 │ tooltipActions: [

106 │ // Nút Skip chỉ hiện khi KHÔNG phải là thankYouKey

107 │ if (key != ShowcaseKeys.thankYouKey)

108 │ TooltipActionButton(

109 │ type: TooltipDefaultActionType.skip,

110 │ name: ShowcaseText.getSkipButton(context),

111 │ textStyle: const TextStyle(color: Colors.white),

112 │ backgroundColor: Colors.grey,

113 │ onTap: () {

114 │ ShowCaseWidget.of(context).dismiss();

115 │ },

116 │ ),

117 │ // Nút Next/Close với text tùy chỉnh

118 │ if (key != ShowcaseKeys.thankYouKey)

119 │ TooltipActionButton(

120 │ type: TooltipDefaultActionType.next,

121 │ // Nếu là welcomeKey thì hiện "Xem hướng dẫn", ngược lại hiện "Tiếp"

122 │ name: key == ShowcaseKeys.welcomeKey

123 │ ? ShowcaseText.getViewGuideButton(context)

124 │ : ShowcaseText.getNextButton(context),

125 │ textStyle: const TextStyle(color: Colors.white),

126 │ backgroundColor: AppColors.textColorBlue,

127 │ )

128 │ else

129 │ TooltipActionButton(

130 │ type: TooltipDefaultActionType.skip,

131 │ name: ShowcaseText.getCloseButton(context),

132 │ textStyle: const TextStyle(color: Colors.white),

133 │ backgroundColor: AppColors.textColorBlue,

134 │ onTap: () {

135 │ ShowCaseWidget.of(context).dismiss();

136 │ },

137 │ ),

138 │ ],

139 │ child: child,

140 │ );

141 │ }

142 │

143 │ @override

144 │ void initState() {

145 │ super.initState();

146 │ \_checkFirstTime();

147 │ }

148 │

149 │ Future<void> \_resetShowCase() async {

150 │ final prefs = await SharedPreferences.getInstance();

151 │ await prefs.setBool('is\_first\_time', true);

152 │ }

153 │

154 │ Future<void> \_checkFirstTime() async {

155 │ if (!\_enableShowcase) return;

156 │ final prefs = await SharedPreferences.getInstance();

157 │ bool isFirstTime = prefs.getBool('is\_first\_time') ?? true;

158 │

159 │ if (isFirstTime) {

160 │ // Lần đầu mở app

161 │ await prefs.setBool('is\_first\_time', false);

162 │

163 │ // Đảm bảo widget đã được build xong

164 │ WidgetsBinding.instance.addPostFrameCallback((\_) {

165 │ ShowCaseWidget.of(context).startShowCase([

166 │ ShowcaseKeys.welcomeKey,

167 │ ShowcaseKeys.appBarKey,

168 │ ShowcaseKeys.dangKyKhamKey,

169 │ ShowcaseKeys.lichSuKhamKey,

170 │ ShowcaseKeys.tiemChungKey,

171 │ ShowcaseKeys.thamMyKey,

172 │ ShowcaseKeys.newsKey,

173 │ ShowcaseKeys.doctorsKey,

174 │ ShowcaseKeys.taiLieuKey,

175 │ ShowcaseKeys.thankYouKey,

176 │ ]);

177 │ });

178 │ }

179 │ }

180 │

181 │ @override

182 │ Widget build(BuildContext context) {

183 │ return HomeListener(

184 │ child: SeasonalEffectWrapper(

185 │ child: Scaffold(

186 │ appBar: PreferredSize(

187 │ preferredSize: const Size.fromHeight(kToolbarHeight),

188 │ child: \_buildShowcase(

189 │ key: ShowcaseKeys.appBarKey,

190 │ title: ShowcaseText.getSettingsTitle(context),

191 │ description: ShowcaseText.getSettingsDescription(context),

192 │ child: CustomAppBar1(

193 │ context: context,

194 │ ),

195 │ ),

196 │ ),

197 │ drawer: const CustomDrawer(),

198 │ body: DecoratedBox(

199 │ decoration: const BoxDecoration(

200 │ gradient: LinearGradient(

201 │ colors: [AppColors.white, AppColors.colorGreenLight],

202 │ begin: Alignment.topCenter,

203 │ end: Alignment.bottomCenter,

204 │ ),

205 │ ),

206 │ child: RefreshIndicator(

207 │ onRefresh: \_handleRefresh,

208 │ child: SingleChildScrollView(

209 │ physics: const AlwaysScrollableScrollPhysics(),

210 │ child: Column(

211 │ crossAxisAlignment: CrossAxisAlignment.start,

212 │ children: [

213 │ HomeIconsSection(),

214 │ \_buildShowcase(

215 │ key: ShowcaseKeys.welcomeKey,

216 │ title: ShowcaseText.getWelcomeTitle(context),

217 │ targetPadding: EdgeInsets.zero,

218 │ description: ShowcaseText.getWelcomeDescription(context),

219 │ child:

220 │ \_buildPlaceholder(ShowcaseKeys.welcomePlaceholderKey),

221 │ ),

222 │ \_buildShowcase(

223 │ key: ShowcaseKeys.thankYouKey,

224 │ title: ShowcaseText.getThankYouTitle(context),

225 │ targetPadding: EdgeInsets.zero,

226 │ description: ShowcaseText.getThankYouDescription(context),

227 │ enableAutoScroll: true,

228 │ child: \_buildPlaceholder(

229 │ ShowcaseKeys.thankYouPlaceholderKey),

230 │ ),

231 │ \_buildShowcase(

232 │ key: ShowcaseKeys.newsKey,

233 │ title: ShowcaseText.getNewsTitle(context),

234 │ description: ShowcaseText.getNewsDescription(context),

235 │ child: const NewsListSection(),

236 │ ),

237 │ \_buildShowcase(

238 │ key: ShowcaseKeys.doctorsKey,

239 │ title: ShowcaseText.getDoctorsTitle(context),

240 │ description: ShowcaseText.getDoctorsDescription(context),

241 │ child: const DoctorsListSection(),

242 │ ),

243 │ const SizedBox(

244 │ height: AppDimensions.sizeBoxM,

245 │ ),

246 │ \_buildShowcase(

247 │ key: ShowcaseKeys.taiLieuKey,

248 │ title: ShowcaseText.getDocumentsTitle(context),

249 │ description: ShowcaseText.getCosmeticDescription(context),

250 │ enableAutoScroll: true,

251 │ child: const TaiLieuSongKhoeSection(),

252 │ ),

253 │ const AdsListSection(),

254 │ const SizedBox(

255 │ height: AppDimensions.sizeBoxXL,

256 │ )

257 │ ],

258 │ ),

259 │ ),

260 │ ),

261 │ ),

262 │ ),

263 │ ),

264 │ );

265 │ }

266 │ }

────────────────────────────────────────────────────────────

**📄 2\_home\_listener.dart (DART, 62 lines)**

1 │ import 'package:dkkb\_app/4\_presention/bloc/current\_user/current\_user\_bloc.dart';

2 │ import 'package:dkkb\_app/4\_presention/bloc/home/home\_bloc.dart';

3 │ import 'package:dkkb\_app/8\_service/dialog\_service/birthday\_dialog.dart';

4 │ import 'package:dkkb\_app/8\_service/dialog\_service/dialog\_service.dart';

5 │ import 'package:dkkb\_app/commons.dart';

6 │ import 'package:flutter/material.dart';

7 │

8 │ class HomeListener extends StatelessWidget {

9 │ final Widget child;

10 │ const HomeListener({

11 │ super.key,

12 │ required this.child,

13 │ });

14 │

15 │ @override

16 │ Widget build(BuildContext context) {

17 │ // Sử dụng MultiBlocListener để lắng nghe nhiều Bloc

18 │ return MultiBlocListener(

19 │ listeners: [

20 │ // Listener hiện có cho HomeBloc

21 │ BlocListener<HomeBloc, HomeState>(

22 │ listener: (context, state) {

23 │ switch (state.status) {

24 │ case HomePageStatus.failure:

25 │ if (state.errorMessage != null) {

26 │ DialogService.showErrorDialog(context, state.errorMessage!);

27 │ }

28 │ break;

29 │ case HomePageStatus.success:

30 │ // Xử lý khi tải dữ liệu thành công, nếu cần

31 │ break;

32 │ case HomePageStatus.loading:

33 │ // Có thể hiển thị loading indicator nếu cần

34 │ break;

35 │ case HomePageStatus.initial:

36 │ // Xử lý trạng thái ban đầu nếu cần

37 │ break;

38 │ }

39 │ },

40 │ ),

41 │

42 │ // Thêm listener mới cho CurrentUserBloc

43 │ BlocListener<CurrentUserBloc, CurrentUserState>(

44 │ listener: (context, state) {

45 │ state.maybeWhen(

46 │ showBirthdayDialog: (userName) {

47 │ showDialog(

48 │ context: context,

49 │ barrierDismissible:

50 │ false, // Người dùng phải tương tác với dialog

51 │ builder: (context) => BirthdayDialog(userName: userName),

52 │ );

53 │ },

54 │ orElse: () {},

55 │ );

56 │ },

57 │ ),

58 │ ],

59 │ child: child,

60 │ );

61 │ }

62 │ }

────────────────────────────────────────────────────────────

**📄 3a\_news\_list\_section.dart (DART, 452 lines)**

1 │ // ignore\_for\_file: deprecated\_member\_use

2 │

3 │ import 'package:auto\_size\_text/auto\_size\_text.dart';

4 │ import 'package:cached\_network\_image/cached\_network\_image.dart';

5 │ import 'package:dkkb\_app/4\_presention/bloc/home\_bottombar\_navigator/home\_bottombar\_navigator\_bloc.dart';

6 │ import 'package:dkkb\_app/4\_presention/cubit/remember\_me/unread\_news\_cubit.dart';

7 │ import 'package:dkkb\_app/4\_presention/widget/badge/notification\_badge.dart';

8 │ import 'package:dkkb\_app/5\_app/app.dart';

9 │ //import 'package:dkkb\_app/8\_service/in\_app\_webview/base\_web\_view.dart';

10 │ import 'package:dkkb\_app/7\_core/responsive/media\_query\_extension.dart';

11 │ import 'package:dkkb\_app/commons.dart';

12 │ import 'package:flutter/material.dart';

13 │ import 'package:flutter\_cache\_manager/flutter\_cache\_manager.dart';

14 │ import 'package:shimmer/shimmer.dart';

15 │ import 'package:dkkb\_app/1\_models/news/news\_model.dart';

16 │ import 'package:dkkb\_app/4\_presention/bloc/home/home\_bloc.dart';

17 │ import 'package:url\_launcher/url\_launcher.dart';

18 │

19 │ class NewsListSection extends StatelessWidget {

20 │ final String title;

21 │ final int itemCount;

22 │

23 │ const NewsListSection({

24 │ super.key,

25 │ this.title = "Tin tức",

26 │ this.itemCount = 5,

27 │ });

28 │

29 │ @override

30 │ Widget build(BuildContext context) {

31 │ return BlocBuilder<HomeBloc, HomeState>(

32 │ buildWhen: (previous, current) =>

33 │ previous.news != current.news || previous.status != current.status,

34 │ builder: (context, state) {

35 │ return Column(

36 │ crossAxisAlignment: CrossAxisAlignment.start,

37 │ children: [

38 │ Row(

39 │ mainAxisAlignment: MainAxisAlignment.spaceBetween,

40 │ children: [

41 │ Padding(

42 │ padding: EdgeInsets.only(left: 10.w),

43 │ child: AutoSizeText(

44 │ AppLocalizations.of(context).home\_tintuc,

45 │ style: Theme.of(context).textTheme.bodyMedium!.copyWith(

46 │ fontWeight: FontWeight.bold,

47 │ color: AppColors.textColorGreen,

48 │ fontSize: 20,

49 │ ),

50 │ ),

51 │ ),

52 │ TextButton(

53 │ onPressed: () {

54 │ context

55 │ .read<HomeNavigationBloc>()

56 │ .add(HomeNavigateToIndex(3));

57 │ },

58 │ style: TextButton.styleFrom(

59 │ foregroundColor: AppColors.blueHis2, // Màu của text

60 │ padding: const EdgeInsets.symmetric(

61 │ horizontal: 20, vertical: 10),

62 │ shape: RoundedRectangleBorder(

63 │ borderRadius: BorderRadius.circular(25.r),

64 │ ),

65 │ ),

66 │ child: Text(

67 │ AppLocalizations.of(context).mainScreen\_xemtoanbo,

68 │ style: Theme.of(context).textTheme.bodySmall!.copyWith(

69 │ color: AppColors.blueHis2,

70 │ fontStyle: FontStyle.italic,

71 │ ),

72 │ ),

73 │ ),

74 │ ],

75 │ ),

76 │ StreamBuilder<List<NewsModel>>(

77 │ stream: context.read<HomeBloc>().newsStream,

78 │ builder: (context, snapshot) {

79 │ if (snapshot.connectionState == ConnectionState.waiting) {

80 │ return \_buildShimmerLoading();

81 │ } else if (snapshot.hasError) {

82 │ return \_buildErrorState(context, snapshot.error.toString());

83 │ } else if (!snapshot.hasData || snapshot.data!.isEmpty) {

84 │ return \_buildEmptyState(context);

85 │ } else {

86 │ return \_buildNewsList(context, snapshot.data!);

87 │ }

88 │ },

89 │ ),

90 │ ],

91 │ );

92 │ },

93 │ );

94 │ }

95 │

96 │ Widget \_buildShimmerLoading() {

97 │ return Shimmer.fromColors(

98 │ baseColor: Colors.grey[300]!,

99 │ highlightColor: Colors.grey[100]!,

100 │ child: SizedBox(

101 │ height: 200.h,

102 │ child: ListView.builder(

103 │ scrollDirection: Axis.horizontal,

104 │ itemCount: itemCount,

105 │ itemBuilder: (context, index) => Container(

106 │ width: 300.w,

107 │ margin: EdgeInsets.symmetric(horizontal: 8.w, vertical: 5.h),

108 │ decoration: BoxDecoration(

109 │ color: Colors.white,

110 │ borderRadius: BorderRadius.circular(15.r),

111 │ ),

112 │ ),

113 │ ),

114 │ ),

115 │ );

116 │ }

117 │

118 │ Widget \_buildErrorState(BuildContext context, String? errorMessage) {

119 │ return Container(

120 │ height: 200.h,

121 │ margin: EdgeInsets.symmetric(horizontal: 15.w),

122 │ decoration: BoxDecoration(

123 │ color: Colors.white,

124 │ borderRadius: BorderRadius.circular(15.r),

125 │ boxShadow: [

126 │ BoxShadow(

127 │ color: Colors.grey.withValues(alpha: 0.3),

128 │ spreadRadius: 1,

129 │ blurRadius: 3,

130 │ offset: const Offset(0, 2),

131 │ ),

132 │ ],

133 │ ),

134 │ child: Column(

135 │ mainAxisAlignment: MainAxisAlignment.center,

136 │ children: [

137 │ Icon(

138 │ Icons.error\_outline,

139 │ size: 40.sp,

140 │ color: Colors.red[300],

141 │ ),

142 │ SizedBox(height: 10.h),

143 │ Text(

144 │ errorMessage ?? "Đã xảy ra lỗi khi tải tin tức",

145 │ style: Theme.of(context).textTheme.bodyMedium?.copyWith(

146 │ color: Colors.grey[600],

147 │ ),

148 │ textAlign: TextAlign.center,

149 │ ),

150 │ SizedBox(height: 15.h),

151 │ ElevatedButton.icon(

152 │ onPressed: () {

153 │ context

154 │ .read<HomeBloc>()

155 │ .add(const HomeEvent.started(forceRefresh: true));

156 │ },

157 │ icon: const Icon(Icons.refresh),

158 │ label: const Text("Thử lại"),

159 │ style: ElevatedButton.styleFrom(

160 │ backgroundColor: AppColors.blueHis2,

161 │ foregroundColor: Colors.white,

162 │ padding: EdgeInsets.symmetric(horizontal: 20.w, vertical: 10.h),

163 │ shape: RoundedRectangleBorder(

164 │ borderRadius: BorderRadius.circular(20.r),

165 │ ),

166 │ ),

167 │ ),

168 │ ],

169 │ ),

170 │ );

171 │ }

172 │

173 │ Widget \_buildEmptyState(BuildContext context) {

174 │ return Container(

175 │ height: 200.h,

176 │ width: context.percentWidth(80),

177 │ margin: EdgeInsets.symmetric(horizontal: 15.w),

178 │ decoration: BoxDecoration(

179 │ color: Colors.white,

180 │ borderRadius: BorderRadius.circular(15.r),

181 │ boxShadow: [

182 │ BoxShadow(

183 │ color: Colors.grey.withValues(alpha: 0.3),

184 │ spreadRadius: 1,

185 │ blurRadius: 3,

186 │ offset: const Offset(0, 2),

187 │ ),

188 │ ],

189 │ ),

190 │ child: Column(

191 │ mainAxisAlignment: MainAxisAlignment.center,

192 │ children: [

193 │ Icon(

194 │ Icons.article\_outlined,

195 │ size: 40.sp,

196 │ color: Colors.grey[400],

197 │ ),

198 │ SizedBox(height: 10.h),

199 │ Text(

200 │ "Không có tin tức mới",

201 │ style: Theme.of(context).textTheme.bodyMedium?.copyWith(

202 │ color: Colors.grey[600],

203 │ ),

204 │ textAlign: TextAlign.center,

205 │ ),

206 │ SizedBox(height: 15.h),

207 │ ElevatedButton.icon(

208 │ onPressed: () {

209 │ context

210 │ .read<HomeBloc>()

211 │ .add(const HomeEvent.started(forceRefresh: true));

212 │ },

213 │ icon: const Icon(Icons.refresh),

214 │ label: const Text("Thử lại"),

215 │ style: ElevatedButton.styleFrom(

216 │ backgroundColor: AppColors.blueHis2,

217 │ foregroundColor: Colors.white,

218 │ padding: EdgeInsets.symmetric(horizontal: 20.w, vertical: 10.h),

219 │ shape: RoundedRectangleBorder(

220 │ borderRadius: BorderRadius.circular(20.r),

221 │ ),

222 │ ),

223 │ ),

224 │ ],

225 │ ),

226 │ );

227 │ }

228 │

229 │ Widget \_buildNewsList(BuildContext context, List<NewsModel> news) {

230 │ final displayList = news.take(itemCount).toList();

231 │ return SizedBox(

232 │ height: 160.h,

233 │ child: ListView.builder(

234 │ padding: EdgeInsets.symmetric(horizontal: 8.w),

235 │ scrollDirection: Axis.horizontal,

236 │ itemCount: displayList.length,

237 │ itemBuilder: (context, index) {

238 │ final newsItem = displayList[index];

239 │ return \_NewsCard(news: newsItem);

240 │ },

241 │ ),

242 │ );

243 │ }

244 │ }

245 │

246 │ class \_NewsCard extends StatelessWidget {

247 │ final NewsModel news;

248 │

249 │ const \_NewsCard({required this.news});

250 │

251 │ bool isValidImageUrl(String url) {

252 │ return url.isNotEmpty && Uri.tryParse(url)?.hasAbsolutePath == true;

253 │ }

254 │

255 │ Future<void> \_launchUrl(Uri url) async {

256 │ try {

257 │ await launchUrl(url, mode: LaunchMode.externalApplication);

258 │ } catch (e) {

259 │ AppLogger.error('Error launching URL: $e');

260 │ throw Exception('Could not launch $url');

261 │ }

262 │ }

263 │

264 │ Future<void> \_handleTap(BuildContext context) async {

265 │ final uri = Uri.tryParse(news.url);

266 │ if (uri != null) {

267 │ var connectivityResult = await Connectivity().checkConnectivity();

268 │ if (connectivityResult != ConnectivityResult.none) {

269 │ try {

270 │ final launched =

271 │ await launchUrl(uri, mode: LaunchMode.externalApplication);

272 │ if (launched) {

273 │ WidgetsBinding.instance.addPostFrameCallback(

274 │ (\_) {

275 │ context.read<NewsCountCubit>().markNewsAsRead(news.ma);

276 │ },

277 │ );

278 │ } else {

279 │ if (!context.mounted) return;

280 │ DialogService.showErrorDialog(

281 │ context,

282 │ "Không thể mở ứng dụng\nVui lòng thử lại sau",

283 │ );

284 │ }

285 │ } catch (e) {

286 │ if (!context.mounted) return;

287 │ DialogService.showErrorDialog(

288 │ context,

289 │ "Đã xảy ra lỗi\nVui lòng thử lại sau",

290 │ );

291 │ }

292 │ } else {

293 │ if (!context.mounted) return;

294 │ DialogService.showErrorDialog(

295 │ context,

296 │ "Không thể mở nội dung\nVui lòng kiểm tra kết nối mạng",

297 │ );

298 │ }

299 │ } else {

300 │ if (!context.mounted) return;

301 │ DialogService.showErrorDialog(

302 │ context,

303 │ "Đường dẫn không hợp lệ\nVui lòng thử lại sau",

304 │ );

305 │ }

306 │ }

307 │

308 │ @override

309 │ Widget build(BuildContext context) {

310 │ return InkWell(

311 │ onTap: () => \_handleTap(context),

312 │ child: Container(

313 │ width: context.percentWidth(85),

314 │ height: context.percentWidth(80) / 1.91,

315 │ margin: EdgeInsets.symmetric(

316 │ horizontal: context.responsiveWidth(2),

317 │ vertical: context.responsiveHeight(1),

318 │ ),

319 │ decoration: BoxDecoration(

320 │ borderRadius: BorderRadius.circular(context.responsiveRadius(4)),

321 │ color: Colors.white,

322 │ boxShadow: [

323 │ BoxShadow(

324 │ color: Colors.grey.withValues(alpha: 0.5),

325 │ spreadRadius: 1,

326 │ blurRadius: 2,

327 │ offset: const Offset(-1, 3),

328 │ ),

329 │ ],

330 │ ),

331 │ child: ClipRRect(

332 │ borderRadius: BorderRadius.circular(context.responsiveRadius(4)),

333 │ child: BlocBuilder<NewsCountCubit, NewsCountState>(

334 │ builder: (context, state) {

335 │ final isUnread =

336 │ !context.read<NewsCountCubit>().isNewsRead(news.ma);

337 │ return NotificationBadge(

338 │ isVisible: isUnread && state.showBadge,

339 │ child: isValidImageUrl(news.hinhAnh)

340 │ ? CachedNetworkImage(

341 │ imageUrl: news.hinhAnh,

342 │ width: double.infinity,

343 │ height: double.infinity,

344 │ fit: BoxFit.fill,

345 │ placeholder: (context, url) => Shimmer.fromColors(

346 │ baseColor: Colors.grey[300]!,

347 │ highlightColor: Colors.grey[100]!,

348 │ child: Container(

349 │ color: Colors.white,

350 │ ),

351 │ ),

352 │ errorWidget: (context, url, error) => Container(

353 │ color: Colors.grey[300],

354 │ child: Column(

355 │ mainAxisAlignment: MainAxisAlignment.center,

356 │ children: [

357 │ Icon(

358 │ Icons.broken\_image,

359 │ size: context.responsiveWidth(10),

360 │ color: Colors.grey[500],

361 │ ),

362 │ const SizedBox(height: 4),

363 │ Text(

364 │ 'Không thể tải ảnh',

365 │ style: TextStyle(

366 │ fontSize: context.scale(10),

367 │ color: Colors.grey[600],

368 │ ),

369 │ ),

370 │ const SizedBox(height: 8),

371 │ ElevatedButton.icon(

372 │ onPressed: () {

373 │ DefaultCacheManager()

374 │ .removeFile(url)

375 │ .then((\_) {

376 │ CachedNetworkImage.evictFromCache(url);

377 │ if (context.mounted) {

378 │ (context as Element).markNeedsBuild();

379 │ }

380 │ });

381 │ },

382 │ icon: Icon(

383 │ Icons.refresh,

384 │ size: context.responsiveWidth(5),

385 │ ),

386 │ label: Text(

387 │ 'Tải lại',

388 │ style: TextStyle(

389 │ fontSize: context.scale(10),

390 │ ),

391 │ ),

392 │ style: ElevatedButton.styleFrom(

393 │ backgroundColor: AppColors.blueHis2,

394 │ foregroundColor: Colors.white,

395 │ padding: EdgeInsets.symmetric(

396 │ horizontal: context.responsiveWidth(4),

397 │ vertical: context.responsiveHeight(1),

398 │ ),

399 │ shape: RoundedRectangleBorder(

400 │ borderRadius: BorderRadius.circular(

401 │ context.responsiveRadius(2)),

402 │ ),

403 │ ),

404 │ ),

405 │ ],

406 │ ),

407 │ ),

408 │ cacheManager: DefaultCacheManager(),

409 │ maxHeightDiskCache: 1080,

410 │ memCacheHeight: 1080,

411 │ )

412 │ : Container(

413 │ color: Colors.grey[300],

414 │ child: Column(

415 │ mainAxisAlignment: MainAxisAlignment.center,

416 │ children: [

417 │ Icon(

418 │ Icons.broken\_image,

419 │ size: context.responsiveWidth(10),

420 │ color: Colors.grey[500],

421 │ ),

422 │ const SizedBox(height: 8),

423 │ Padding(

424 │ padding: EdgeInsets.symmetric(

425 │ horizontal: context.responsiveWidth(2),

426 │ ),

427 │ child: Text(

428 │ news.ten,

429 │ maxLines: 2,

430 │ textAlign: TextAlign.center,

431 │ overflow: TextOverflow.ellipsis,

432 │ style: Theme.of(context)

433 │ .textTheme

434 │ .bodyMedium

435 │ ?.copyWith(

436 │ fontWeight: FontWeight.bold,

437 │ fontSize: context.scale(12.sp),

438 │ color: AppColors.textColorBlue,

439 │ ),

440 │ ),

441 │ ),

442 │ ],

443 │ ),

444 │ ),

445 │ );

446 │ },

447 │ ),

448 │ ),

449 │ ),

450 │ );

451 │ }

452 │ }

────────────────────────────────────────────────────────────

**📄 3b\_doctors\_section.dart (DART, 428 lines)**

1 │ // ignore\_for\_file: unused\_field

2 │

3 │ import 'package:cached\_network\_image/cached\_network\_image.dart';

4 │ import 'package:dkkb\_app/1\_models/doctor/strategies/bac\_si\_list/bac\_si\_list\_model.dart';

5 │ import 'package:dkkb\_app/4\_presention/bloc/home/home\_bloc.dart';

6 │ import 'package:dkkb\_app/4\_presention/bloc/home\_bottombar\_navigator/home\_bottombar\_navigator\_bloc.dart';

7 │ import 'package:dkkb\_app/8\_service/dialog\_service/dialog\_service.dart';

8 │ //import 'package:dkkb\_app/8\_service/in\_app\_webview/base\_web\_view.dart';

9 │ import 'package:dkkb\_app/7\_core/responsive/media\_query\_extension.dart';

10 │ import 'package:dkkb\_app/commons.dart';

11 │ import 'package:flutter/material.dart';

12 │ import 'package:dkkb\_app/5\_app/color/app\_colors.dart';

13 │ import 'package:shimmer/shimmer.dart';

14 │ import 'package:url\_launcher/url\_launcher.dart';

15 │

16 │ class DoctorsListSection extends StatefulWidget {

17 │ final String title;

18 │ final int itemCount;

19 │

20 │ const DoctorsListSection({

21 │ super.key,

22 │ this.title = "Bác sĩ",

23 │ this.itemCount = 5,

24 │ });

25 │

26 │ @override

27 │ State<DoctorsListSection> createState() => \_DoctorsListSectionState();

28 │ }

29 │

30 │ class \_DoctorsListSectionState extends State<DoctorsListSection>

31 │ with SingleTickerProviderStateMixin {

32 │ final ScrollController \_scrollController = ScrollController();

33 │ List<BacSiListModel> displayList = [];

34 │ late AnimationController \_animationController;

35 │ late Animation<double> \_fadeAnimation;

36 │

37 │ @override

38 │ void initState() {

39 │ super.initState();

40 │ \_scrollController.addListener(\_scrollListener);

41 │

42 │ // Khởi tạo animation controller

43 │ \_animationController = AnimationController(

44 │ vsync: this,

45 │ duration: const Duration(milliseconds: 500),

46 │ );

47 │

48 │ // Tạo fade animation

49 │ \_fadeAnimation = Tween<double>(begin: 1.0, end: 0.0).animate(

50 │ CurvedAnimation(

51 │ parent: \_animationController,

52 │ curve: Curves.easeInOut,

53 │ ),

54 │ );

55 │ }

56 │

57 │ @override

58 │ void dispose() {

59 │ \_scrollController.dispose();

60 │ \_animationController.dispose();

61 │ super.dispose();

62 │ }

63 │

64 │ void \_scrollListener() {

65 │ if (\_scrollController.position.pixels ==

66 │ \_scrollController.position.maxScrollExtent) {

67 │ \_refreshDisplayListWithAnimation();

68 │ }

69 │ }

70 │

71 │ Future<void> \_refreshDisplayListWithAnimation() async {

72 │ // Bắt đầu animation fade out

73 │ await \_animationController.forward();

74 │

75 │ // Cập nhật danh sách

76 │ setState(() {

77 │ final doctors = context.read<HomeBloc>().state.doctors;

78 │ final shuffledList = List<BacSiListModel>.from(doctors);

79 │ shuffledList.shuffle();

80 │ displayList = shuffledList.take(widget.itemCount).toList();

81 │ });

82 │

83 │ // Reset và chạy animation fade in

84 │ await \_animationController.reverse();

85 │ }

86 │

87 │ @override

88 │ Widget build(BuildContext context) {

89 │ return BlocBuilder<HomeBloc, HomeState>(

90 │ buildWhen: (previous, current) =>

91 │ previous.doctors != current.doctors ||

92 │ previous.status != current.status,

93 │ builder: (context, state) {

94 │ return Column(

95 │ crossAxisAlignment: CrossAxisAlignment.start,

96 │ children: [

97 │ Padding(

98 │ padding: EdgeInsets.only(left: 15.w, right: 10.w),

99 │ child: Row(

100 │ mainAxisAlignment: MainAxisAlignment.spaceBetween,

101 │ children: [

102 │ Text(

103 │ AppLocalizations.of(context).mainScreen\_bacsi,

104 │ style: Theme.of(context).textTheme.bodyMedium!.copyWith(

105 │ fontWeight: FontWeight.bold,

106 │ color: AppColors.textColorGreen,

107 │ fontSize: 20),

108 │ ),

109 │ Padding(

110 │ padding: EdgeInsets.only(top: 5.h),

111 │ child: TextButton(

112 │ onPressed: () {

113 │ context

114 │ .read<HomeNavigationBloc>()

115 │ .add(HomeNavigateToIndex(1));

116 │ },

117 │ child: Text(

118 │ AppLocalizations.of(context).mainScreen\_xemtoanbo,

119 │ style: Theme.of(context).textTheme.bodySmall!.copyWith(

120 │ color: AppColors.blueHis2,

121 │ fontStyle: FontStyle.italic,

122 │ ),

123 │ ),

124 │ ),

125 │ ),

126 │ ],

127 │ ),

128 │ ),

129 │ StreamBuilder<List<BacSiListModel>>(

130 │ stream: context.read<HomeBloc>().doctorsStream,

131 │ builder: (context, snapshot) {

132 │ if (snapshot.connectionState == ConnectionState.waiting) {

133 │ return \_buildShimmerLoading();

134 │ } else if (snapshot.hasError) {

135 │ return \_buildErrorState(context, snapshot.error.toString());

136 │ } else if (!snapshot.hasData || snapshot.data!.isEmpty) {

137 │ return \_buildEmptyState(context);

138 │ } else {

139 │ return \_buildDoctorsList(context, snapshot.data!);

140 │ }

141 │ },

142 │ ),

143 │ ],

144 │ );

145 │ },

146 │ );

147 │ }

148 │

149 │ Widget \_buildDoctorsList(BuildContext context, List<BacSiListModel> doctors) {

150 │ // Khởi tạo displayList nếu rỗng

151 │ if (displayList.isEmpty) {

152 │ final shuffledList = List<BacSiListModel>.from(doctors);

153 │ shuffledList.shuffle();

154 │ displayList = shuffledList.take(widget.itemCount).toList();

155 │ }

156 │

157 │ return SizedBox(

158 │ height: 265.h,

159 │ child: FadeTransition(

160 │ opacity:

161 │ Tween<double>(begin: 1.0, end: 0.0).animate(\_animationController),

162 │ child: ListView.builder(

163 │ controller: \_scrollController,

164 │ padding: EdgeInsets.symmetric(horizontal: 8.w),

165 │ scrollDirection: Axis.horizontal,

166 │ itemCount: displayList.length,

167 │ itemBuilder: (context, index) {

168 │ final doctor = displayList[index];

169 │ return \_DoctorCard(doctor: doctor);

170 │ },

171 │ ),

172 │ ),

173 │ );

174 │ }

175 │

176 │ Widget \_buildShimmerLoading() {

177 │ return Shimmer.fromColors(

178 │ baseColor: Colors.grey[300]!,

179 │ highlightColor: Colors.grey[100]!,

180 │ child: SizedBox(

181 │ height: 260.h,

182 │ child: ListView.builder(

183 │ scrollDirection: Axis.horizontal,

184 │ itemCount: widget.itemCount,

185 │ itemBuilder: (context, index) => Container(

186 │ width: 200.w,

187 │ margin: EdgeInsets.symmetric(horizontal: 8.w, vertical: 5.h),

188 │ decoration: BoxDecoration(

189 │ color: Colors.white,

190 │ borderRadius: BorderRadius.circular(15.r),

191 │ ),

192 │ ),

193 │ ),

194 │ ),

195 │ );

196 │ }

197 │

198 │ Widget \_buildErrorState(BuildContext context, String? errorMessage) {

199 │ return Center(

200 │ child: Column(

201 │ mainAxisAlignment: MainAxisAlignment.center,

202 │ children: [

203 │ Text(

204 │ errorMessage ?? "Đã xảy ra lỗi khi tải danh sách bác sĩ",

205 │ style: Theme.of(context).textTheme.bodyMedium,

206 │ ),

207 │ SizedBox(height: 10.h),

208 │ ElevatedButton(

209 │ onPressed: () {

210 │ context

211 │ .read<HomeBloc>()

212 │ .add(const HomeEvent.started(forceRefresh: true));

213 │ },

214 │ child: const Text('Thử lại'),

215 │ ),

216 │ ],

217 │ ),

218 │ );

219 │ }

220 │

221 │ Widget \_buildEmptyState(BuildContext context) {

222 │ return Center(

223 │ child: Column(

224 │ mainAxisAlignment: MainAxisAlignment.center,

225 │ children: [

226 │ Text(

227 │ "Không có bác sĩ nào",

228 │ style: Theme.of(context).textTheme.bodyMedium,

229 │ ),

230 │ SizedBox(height: 10.h),

231 │ ElevatedButton(

232 │ onPressed: () {

233 │ context

234 │ .read<HomeBloc>()

235 │ .add(const HomeEvent.started(forceRefresh: true));

236 │ },

237 │ child: const Text('Thử lại'),

238 │ ),

239 │ ],

240 │ ),

241 │ );

242 │ }

243 │ }

244 │

245 │ class \_DoctorCard extends StatelessWidget {

246 │ final BacSiListModel doctor;

247 │

248 │ const \_DoctorCard({required this.doctor});

249 │

250 │ @override

251 │ Widget build(BuildContext context) {

252 │ return Container(

253 │ width: context.percentWidth(50),

254 │ margin: context.responsivePadding(horizontal: 2, vertical: 1),

255 │ decoration: BoxDecoration(

256 │ borderRadius: BorderRadius.circular(context.responsiveRadius(4)),

257 │ color: AppColors.textColorBlue,

258 │ boxShadow: [

259 │ BoxShadow(

260 │ color: Colors.black.withValues(alpha: 0.2),

261 │ spreadRadius: 1,

262 │ blurRadius: 2,

263 │ offset: const Offset(-1, 3),

264 │ ),

265 │ ],

266 │ ),

267 │ child: Column(

268 │ children: [

269 │ InkWell(

270 │ onTap: () => \_navigateToDetailScreen(context),

271 │ child: Padding(

272 │ padding: context.responsivePaddingAll(2),

273 │ child: ClipRRect(

274 │ borderRadius:

275 │ BorderRadius.circular(context.responsiveRadius(3)),

276 │ child: CachedNetworkImage(

277 │ imageUrl: doctor.hinhAnh,

278 │ height: context.percentHeight(30),

279 │ width: double.infinity,

280 │ fit: BoxFit.cover,

281 │ alignment: Alignment.topCenter,

282 │ placeholder: (context, url) => Shimmer.fromColors(

283 │ baseColor: Colors.grey[300]!,

284 │ highlightColor: Colors.grey[100]!,

285 │ child: Container(

286 │ color: Colors.white,

287 │ ),

288 │ ),

289 │ errorWidget: (context, url, error) =>

290 │ \_buildErrorImage(context),

291 │ ),

292 │ ),

293 │ ),

294 │ ),

295 │ Expanded(

296 │ child: Center(

297 │ child: Padding(

298 │ padding: context.responsivePadding(horizontal: 2),

299 │ child: Column(

300 │ children: [

301 │ Text(

302 │ doctor.ten,

303 │ textAlign: TextAlign.center,

304 │ maxLines: 2,

305 │ overflow: TextOverflow.ellipsis,

306 │ style: Theme.of(context).textTheme.bodySmall?.copyWith(

307 │ fontWeight: FontWeight.bold,

308 │ color: AppColors.white,

309 │ fontSize: context.scale(12),

310 │ ),

311 │ ),

312 │ Text(

313 │ doctor.tenchucdanh,

314 │ textAlign: TextAlign.center,

315 │ maxLines: 2,

316 │ overflow: TextOverflow.ellipsis,

317 │ style: Theme.of(context).textTheme.bodySmall?.copyWith(

318 │ fontWeight: FontWeight.bold,

319 │ color: AppColors.white,

320 │ fontSize: context.scale(10),

321 │ ),

322 │ ),

323 │ ],

324 │ ),

325 │ ),

326 │ ),

327 │ ),

328 │ ],

329 │ ),

330 │ );

331 │ }

332 │

333 │ Future<void> \_launchUrl(Uri url) async {

334 │ try {

335 │ await launchUrl(url, mode: LaunchMode.externalApplication);

336 │ } catch (e) {

337 │ AppLogger.error('Error launching URL: $e');

338 │ throw Exception('Could not launch $url');

339 │ }

340 │ }

341 │

342 │ Future<void> \_navigateToDetailScreen(BuildContext context) async {

343 │ if (doctor.linkBacSi.isEmpty) return;

344 │

345 │ final uri = Uri.tryParse(doctor.linkBacSi);

346 │ if (uri != null) {

347 │ var connectivityResult = await Connectivity().checkConnectivity();

348 │ if (connectivityResult != ConnectivityResult.none) {

349 │ try {

350 │ final launched =

351 │ await launchUrl(uri, mode: LaunchMode.externalApplication);

352 │ if (!launched) {

353 │ if (!context.mounted) return;

354 │ DialogService.showErrorDialog(

355 │ context,

356 │ "Không thể mở ứng dụng\nVui lòng thử lại sau",

357 │ );

358 │ }

359 │ // Additional actions after successful launch can go here

360 │ } catch (e) {

361 │ if (!context.mounted) return;

362 │ DialogService.showErrorDialog(

363 │ context,

364 │ "Đã xảy ra lỗi\nVui lòng thử lại sau",

365 │ );

366 │ }

367 │ } else {

368 │ if (!context.mounted) return;

369 │ DialogService.showErrorDialog(

370 │ context,

371 │ "Không thể mở nội dung\nVui lòng kiểm tra kết nối mạng",

372 │ );

373 │ }

374 │ } else {

375 │ if (!context.mounted) return;

376 │ DialogService.showErrorDialog(

377 │ context,

378 │ "Đường dẫn không hợp lệ\nVui lòng thử lại sau",

379 │ );

380 │ }

381 │ }

382 │

383 │ Widget \_buildErrorImage(BuildContext context) {

384 │ return Container(

385 │ height: context.percentHeight(25),

386 │ width: double.infinity,

387 │ color: Colors.grey[200],

388 │ child: Column(

389 │ mainAxisAlignment: MainAxisAlignment.center,

390 │ children: [

391 │ Icon(Icons.broken\_image,

392 │ size: context.responsiveWidth(10), color: Colors.grey[400]),

393 │ SizedBox(height: context.responsiveHeight(1)),

394 │ Material(

395 │ color: Colors.transparent,

396 │ child: InkWell(

397 │ onTap: () {

398 │ CachedNetworkImage.evictFromCache(doctor.hinhAnh);

399 │ (context as Element).markNeedsBuild();

400 │ },

401 │ child: Padding(

402 │ padding: context.responsivePadding(horizontal: 2, vertical: 1),

403 │ child: Row(

404 │ mainAxisSize: MainAxisSize.min,

405 │ children: [

406 │ Icon(

407 │ Icons.refresh,

408 │ size: context.responsiveWidth(4),

409 │ color: AppColors.textColorGreen,

410 │ ),

411 │ SizedBox(width: context.responsiveWidth(1)),

412 │ Text(

413 │ 'Tải lại',

414 │ style: TextStyle(

415 │ color: AppColors.textColorGreen,

416 │ fontSize: context.scale(10),

417 │ ),

418 │ ),

419 │ ],

420 │ ),

421 │ ),

422 │ ),

423 │ ),

424 │ ],

425 │ ),

426 │ );

427 │ }

428 │ }

────────────────────────────────────────────────────────────

**📄 3c\_tailieusongkhoe\_section.dart (DART, 436 lines)**

1 │ // ignore\_for\_file: deprecated\_member\_use

2 │

3 │ import 'package:auto\_size\_text/auto\_size\_text.dart';

4 │ import 'package:cached\_network\_image/cached\_network\_image.dart';

5 │ import 'package:dkkb\_app/4\_presention/cubit/remember\_me/unread\_news\_cubit.dart';

6 │ import 'package:dkkb\_app/4\_presention/widget/badge/notification\_badge.dart';

7 │ import 'package:dkkb\_app/5\_app/color/app\_colors.dart';

8 │ import 'package:dkkb\_app/8\_service/dialog\_service/dialog\_service.dart';

9 │ //import 'package:dkkb\_app/8\_service/in\_app\_webview/base\_web\_view.dart';

10 │ import 'package:dkkb\_app/7\_core/responsive/media\_query\_extension.dart';

11 │

12 │ import 'package:dkkb\_app/commons.dart';

13 │ import 'package:flutter/material.dart';

14 │ import 'package:flutter\_cache\_manager/flutter\_cache\_manager.dart';

15 │ import 'package:shimmer/shimmer.dart';

16 │ import 'package:dkkb\_app/1\_models/news/news\_model.dart';

17 │ import 'package:dkkb\_app/4\_presention/bloc/home/home\_bloc.dart';

18 │ import 'package:url\_launcher/url\_launcher.dart';

19 │

20 │ class TaiLieuSongKhoeSection extends StatelessWidget {

21 │ final String title;

22 │ final int itemCount;

23 │

24 │ const TaiLieuSongKhoeSection({

25 │ super.key,

26 │ this.title = "Tài liệu sống khỏe",

27 │ this.itemCount = 100000000000000000,

28 │ });

29 │

30 │ @override

31 │ Widget build(BuildContext context) {

32 │ return BlocBuilder<HomeBloc, HomeState>(

33 │ builder: (context, state) {

34 │ return Column(

35 │ crossAxisAlignment: CrossAxisAlignment.start,

36 │ children: [

37 │ Padding(

38 │ padding: EdgeInsets.only(left: 10.w),

39 │ child: Row(

40 │ mainAxisAlignment: MainAxisAlignment.spaceBetween,

41 │ children: [

42 │ AutoSizeText(

43 │ AppLocalizations.of(context).home\_tailieusongkhoe,

44 │ style: Theme.of(context).textTheme.bodyMedium!.copyWith(

45 │ fontWeight: FontWeight.bold,

46 │ color: AppColors.textColorGreen,

47 │ fontSize: 20,

48 │ ),

49 │ ),

50 │ Padding(

51 │ padding: EdgeInsets.only(top: 5.h),

52 │ child: TextButton(

53 │ onPressed: () {

54 │ \_launchUrlWEB();

55 │ },

56 │ child: Text(

57 │ AppLocalizations.of(context).mainScreen\_xemtoanbo,

58 │ style: Theme.of(context).textTheme.bodySmall!.copyWith(

59 │ color: AppColors.blueHis2,

60 │ fontStyle: FontStyle.italic,

61 │ ),

62 │ ),

63 │ ),

64 │ ),

65 │ ],

66 │ ),

67 │ ),

68 │ StreamBuilder<List<NewsModel>>(

69 │ stream: context.read<HomeBloc>().taiLieuSongKhoeStream,

70 │ builder: (context, snapshot) {

71 │ if (snapshot.connectionState == ConnectionState.waiting) {

72 │ return \_buildShimmerLoading();

73 │ } else if (snapshot.hasError) {

74 │ return \_buildErrorState(context, snapshot.error.toString());

75 │ } else if (!snapshot.hasData || snapshot.data!.isEmpty) {

76 │ return \_buildEmptyState(context);

77 │ } else {

78 │ return \_buildTaiLieuList(context, snapshot.data!);

79 │ }

80 │ },

81 │ ),

82 │ ],

83 │ );

84 │ },

85 │ );

86 │ }

87 │

88 │ Future<void> \_launchUrlWEB() async {

89 │ if (!await launchUrl(

90 │ Uri.parse(

91 │ 'https://honghunghospital.com.vn/category/tin-tuc/ban-tin-y-te/'),

92 │ mode: LaunchMode.externalApplication)) {

93 │ throw Exception('Could not launch web');

94 │ }

95 │ }

96 │

97 │ Widget \_buildShimmerLoading() {

98 │ return Shimmer.fromColors(

99 │ baseColor: Colors.grey[300]!,

100 │ highlightColor: Colors.grey[100]!,

101 │ child: SizedBox(

102 │ height: 200.h,

103 │ child: ListView.builder(

104 │ scrollDirection: Axis.horizontal,

105 │ itemCount: itemCount,

106 │ itemBuilder: (context, index) => Container(

107 │ width: 300.w,

108 │ margin: EdgeInsets.symmetric(horizontal: 8.w, vertical: 5.h),

109 │ decoration: BoxDecoration(

110 │ color: Colors.white,

111 │ borderRadius: BorderRadius.circular(15.r),

112 │ ),

113 │ ),

114 │ ),

115 │ ),

116 │ );

117 │ }

118 │

119 │ Widget \_buildErrorState(BuildContext context, String? errorMessage) {

120 │ return Container(

121 │ height: 200.h,

122 │ margin: EdgeInsets.symmetric(horizontal: 15.w),

123 │ decoration: BoxDecoration(

124 │ color: Colors.white,

125 │ borderRadius: BorderRadius.circular(15.r),

126 │ boxShadow: [

127 │ BoxShadow(

128 │ color: Colors.grey.withValues(alpha: 0.3),

129 │ spreadRadius: 1,

130 │ blurRadius: 3,

131 │ offset: const Offset(0, 2),

132 │ ),

133 │ ],

134 │ ),

135 │ child: Column(

136 │ mainAxisAlignment: MainAxisAlignment.center,

137 │ children: [

138 │ Icon(Icons.error\_outline, size: 40.sp, color: Colors.red[300]),

139 │ SizedBox(height: 10.h),

140 │ Text(

141 │ errorMessage ?? "Đã xảy ra lỗi khi tải tài liệu",

142 │ style: Theme.of(context).textTheme.bodyMedium?.copyWith(

143 │ color: Colors.grey[600],

144 │ ),

145 │ textAlign: TextAlign.center,

146 │ ),

147 │ SizedBox(height: 15.h),

148 │ ElevatedButton.icon(

149 │ onPressed: () {

150 │ context.read<HomeBloc>().add(

151 │ const HomeEvent.started(forceRefresh: true),

152 │ );

153 │ },

154 │ icon: const Icon(Icons.refresh),

155 │ label: const Text("Thử lại"),

156 │ style: ElevatedButton.styleFrom(

157 │ backgroundColor: AppColors.blueHis2,

158 │ foregroundColor: Colors.white,

159 │ padding: EdgeInsets.symmetric(horizontal: 20.w, vertical: 10.h),

160 │ shape: RoundedRectangleBorder(

161 │ borderRadius: BorderRadius.circular(20.r),

162 │ ),

163 │ ),

164 │ ),

165 │ ],

166 │ ),

167 │ );

168 │ }

169 │

170 │ Widget \_buildEmptyState(BuildContext context) {

171 │ return Container(

172 │ height: 180.h,

173 │ width: context.percentWidth(80),

174 │ margin: EdgeInsets.symmetric(horizontal: 15.w),

175 │ decoration: BoxDecoration(

176 │ color: Colors.white,

177 │ borderRadius: BorderRadius.circular(15.r),

178 │ boxShadow: [

179 │ BoxShadow(

180 │ color: Colors.grey.withValues(alpha: 0.3),

181 │ spreadRadius: 1,

182 │ blurRadius: 3,

183 │ offset: const Offset(0, 2),

184 │ ),

185 │ ],

186 │ ),

187 │ child: Column(

188 │ mainAxisAlignment: MainAxisAlignment.center,

189 │ children: [

190 │ Icon(Icons.article\_outlined, size: 40.sp, color: Colors.grey[400]),

191 │ SizedBox(height: 10.h),

192 │ Text(

193 │ "Không có tài liệu mới",

194 │ style: Theme.of(context).textTheme.bodyMedium?.copyWith(

195 │ color: Colors.grey[600],

196 │ ),

197 │ textAlign: TextAlign.center,

198 │ ),

199 │ SizedBox(height: 15.h),

200 │ ElevatedButton.icon(

201 │ onPressed: () {

202 │ context.read<HomeBloc>().add(

203 │ const HomeEvent.started(forceRefresh: true),

204 │ );

205 │ },

206 │ icon: const Icon(Icons.refresh),

207 │ label: const Text("Thử lại"),

208 │ style: ElevatedButton.styleFrom(

209 │ backgroundColor: AppColors.blueHis2,

210 │ foregroundColor: Colors.white,

211 │ padding: EdgeInsets.symmetric(horizontal: 20.w, vertical: 10.h),

212 │ shape: RoundedRectangleBorder(

213 │ borderRadius: BorderRadius.circular(20.r),

214 │ ),

215 │ ),

216 │ ),

217 │ ],

218 │ ),

219 │ );

220 │ }

221 │

222 │ Widget \_buildTaiLieuList(BuildContext context, List<NewsModel> taiLieu) {

223 │ final displayList = taiLieu.take(itemCount).toList();

224 │ return SizedBox(

225 │ height: 160.h,

226 │ child: ListView.builder(

227 │ padding: EdgeInsets.symmetric(horizontal: 8.w),

228 │ scrollDirection: Axis.horizontal,

229 │ itemCount: displayList.length,

230 │ itemBuilder: (context, index) {

231 │ final taiLieuItem = displayList[index];

232 │ return \_TaiLieuCard(taiLieu: taiLieuItem);

233 │ },

234 │ ),

235 │ );

236 │ }

237 │ }

238 │

239 │ class \_TaiLieuCard extends StatelessWidget {

240 │ final NewsModel taiLieu;

241 │

242 │ const \_TaiLieuCard({required this.taiLieu});

243 │

244 │ bool isValidImageUrl(String url) {

245 │ return url.isNotEmpty && Uri.tryParse(url)?.hasAbsolutePath == true;

246 │ }

247 │

248 │ Future<void> \_handleTap(BuildContext context) async {

249 │ final uri = Uri.tryParse(taiLieu.url);

250 │ if (uri != null) {

251 │ var connectivityResult = await Connectivity().checkConnectivity();

252 │ if (connectivityResult != ConnectivityResult.none) {

253 │ try {

254 │ final launched =

255 │ await launchUrl(uri, mode: LaunchMode.externalApplication);

256 │ if (launched) {

257 │ WidgetsBinding.instance.addPostFrameCallback(

258 │ (\_) {

259 │ context.read<NewsCountCubit>().markDocumentAsRead(taiLieu.ma);

260 │ },

261 │ );

262 │ } else {

263 │ if (!context.mounted) return;

264 │ DialogService.showErrorDialog(

265 │ context,

266 │ "Không thể mở ứng dụng\nVui lòng thử lại sau",

267 │ );

268 │ }

269 │ } catch (e) {

270 │ if (!context.mounted) return;

271 │ DialogService.showErrorDialog(

272 │ context,

273 │ "Đã xảy ra lỗi\nVui lòng thử lại sau",

274 │ );

275 │ }

276 │ } else {

277 │ if (!context.mounted) return;

278 │ DialogService.showErrorDialog(

279 │ context,

280 │ "Không thể mở nội dung\nVui lòng kiểm tra kết nối mạng",

281 │ );

282 │ }

283 │ } else {

284 │ if (!context.mounted) return;

285 │ DialogService.showErrorDialog(

286 │ context,

287 │ "Đường dẫn không hợp lệ\nVui lòng thử lại sau",

288 │ );

289 │ }

290 │ }

291 │

292 │ @override

293 │ Widget build(BuildContext context) {

294 │ return InkWell(

295 │ onTap: () => \_handleTap(context),

296 │ child: Container(

297 │ width: context.percentWidth(85),

298 │ height: context.percentWidth(85) / 1.91,

299 │ margin: EdgeInsets.symmetric(

300 │ horizontal: context.responsiveWidth(2),

301 │ vertical: context.responsiveHeight(1),

302 │ ),

303 │ decoration: BoxDecoration(

304 │ borderRadius: BorderRadius.circular(context.responsiveRadius(4)),

305 │ color: Colors.white,

306 │ boxShadow: [

307 │ BoxShadow(

308 │ color: Colors.grey.withValues(alpha: 0.5),

309 │ spreadRadius: 1,

310 │ blurRadius: 2,

311 │ offset: const Offset(-1, 3),

312 │ ),

313 │ ],

314 │ ),

315 │ child: ClipRRect(

316 │ borderRadius: BorderRadius.circular(context.responsiveRadius(4)),

317 │ child: BlocBuilder<NewsCountCubit, NewsCountState>(

318 │ builder: (context, state) {

319 │ final isUnread =

320 │ !context.read<NewsCountCubit>().isDocumentRead(taiLieu.ma);

321 │ return NotificationBadge(

322 │ isVisible: isUnread && state.showBadge,

323 │ child: isValidImageUrl(taiLieu.hinhAnh)

324 │ ? CachedNetworkImage(

325 │ imageUrl: taiLieu.hinhAnh,

326 │ width: double.infinity,

327 │ height: double.infinity,

328 │ fit: BoxFit.fill,

329 │ placeholder: (context, url) => Shimmer.fromColors(

330 │ baseColor: Colors.grey[300]!,

331 │ highlightColor: Colors.grey[100]!,

332 │ child: Container(

333 │ color: Colors.white,

334 │ ),

335 │ ),

336 │ errorWidget: (context, url, error) => Container(

337 │ color: Colors.grey[300],

338 │ child: Column(

339 │ mainAxisAlignment: MainAxisAlignment.center,

340 │ children: [

341 │ Icon(

342 │ Icons.broken\_image,

343 │ size: context.responsiveWidth(10),

344 │ color: Colors.grey[500],

345 │ ),

346 │ const SizedBox(height: 4),

347 │ Text(

348 │ 'Không thể tải ảnh',

349 │ style: TextStyle(

350 │ fontSize: context.scale(10),

351 │ color: Colors.grey[600],

352 │ ),

353 │ ),

354 │ const SizedBox(height: 8),

355 │ ElevatedButton.icon(

356 │ onPressed: () {

357 │ DefaultCacheManager()

358 │ .removeFile(url)

359 │ .then((\_) {

360 │ CachedNetworkImage.evictFromCache(url);

361 │ if (context.mounted) {

362 │ (context as Element).markNeedsBuild();

363 │ }

364 │ });

365 │ },

366 │ icon: Icon(

367 │ Icons.refresh,

368 │ size: context.responsiveWidth(5),

369 │ ),

370 │ label: Text(

371 │ 'Tải lại',

372 │ style: TextStyle(

373 │ fontSize: context.scale(10),

374 │ ),

375 │ ),

376 │ style: ElevatedButton.styleFrom(

377 │ backgroundColor: AppColors.blueHis2,

378 │ foregroundColor: Colors.white,

379 │ padding: EdgeInsets.symmetric(

380 │ horizontal: context.responsiveWidth(4),

381 │ vertical: context.responsiveHeight(1),

382 │ ),

383 │ shape: RoundedRectangleBorder(

384 │ borderRadius: BorderRadius.circular(

385 │ context.responsiveRadius(2)),

386 │ ),

387 │ ),

388 │ ),

389 │ ],

390 │ ),

391 │ ),

392 │ cacheManager: DefaultCacheManager(),

393 │ maxHeightDiskCache: 1080,

394 │ memCacheHeight: 1080,

395 │ )

396 │ : Container(

397 │ color: Colors.grey[300],

398 │ child: Column(

399 │ mainAxisAlignment: MainAxisAlignment.center,

400 │ children: [

401 │ Icon(

402 │ Icons.broken\_image,

403 │ size: context.responsiveWidth(10),

404 │ color: Colors.grey[500],

405 │ ),

406 │ const SizedBox(height: 8),

407 │ Padding(

408 │ padding: EdgeInsets.symmetric(

409 │ horizontal: context.responsiveWidth(2),

410 │ ),

411 │ child: Text(

412 │ taiLieu.ten,

413 │ maxLines: 2,

414 │ textAlign: TextAlign.center,

415 │ overflow: TextOverflow.ellipsis,

416 │ style: Theme.of(context)

417 │ .textTheme

418 │ .bodyMedium

419 │ ?.copyWith(

420 │ fontWeight: FontWeight.bold,

421 │ fontSize: context.scale(12.sp),

422 │ color: AppColors.textColorBlue,

423 │ ),

424 │ ),

425 │ ),

426 │ ],

427 │ ),

428 │ ),

429 │ );

430 │ },

431 │ ),

432 │ ),

433 │ ),

434 │ );

435 │ }

436 │ }

────────────────────────────────────────────────────────────

**📄 3d\_ads\_list\_section.dart (DART, 266 lines)**

1 │ // ignore\_for\_file: deprecated\_member\_use

2 │

3 │ import 'package:dkkb\_app/1\_models/video\_ads/video\_ads\_model.dart';

4 │ import 'package:dkkb\_app/4\_presention/Pages/home/widget/ads\_card/ads\_cards\_widget.dart';

5 │ import 'package:dkkb\_app/generated/l10n/app\_localizations.dart';

6 │ import 'package:flutter/material.dart';

7 │ import 'package:flutter\_bloc/flutter\_bloc.dart';

8 │ import 'package:flutter\_screenutil/flutter\_screenutil.dart';

9 │ import 'package:shimmer/shimmer.dart';

10 │ import 'package:dkkb\_app/4\_presention/bloc/home/home\_bloc.dart';

11 │ import 'package:dkkb\_app/5\_app/color/app\_colors.dart';

12 │ import 'package:url\_launcher/url\_launcher.dart';

13 │

14 │ class AdsListSection extends StatelessWidget {

15 │ final String title;

16 │

17 │ const AdsListSection({

18 │ super.key,

19 │ this.title = "Quảng cáo",

20 │ });

21 │

22 │ @override

23 │ Widget build(BuildContext context) {

24 │ return BlocBuilder<HomeBloc, HomeState>(

25 │ buildWhen: (previous, current) =>

26 │ previous.ads != current.ads || previous.status != current.status,

27 │ builder: (context, state) {

28 │ return Column(

29 │ crossAxisAlignment: CrossAxisAlignment.start,

30 │ children: [

31 │ Padding(

32 │ padding: EdgeInsets.only(left: 15.w, right: 10.w),

33 │ child: Row(

34 │ mainAxisAlignment: MainAxisAlignment.spaceBetween,

35 │ children: [

36 │ Text(

37 │ "Video",

38 │ style: Theme.of(context).textTheme.bodyMedium!.copyWith(

39 │ fontWeight: FontWeight.bold,

40 │ color: AppColors.textColorGreen,

41 │ fontSize: 20,

42 │ ),

43 │ ),

44 │ Padding(

45 │ padding: EdgeInsets.only(top: 5.h),

46 │ child: TextButton(

47 │ onPressed: () {

48 │ \_launchUrlYTB();

49 │ },

50 │ child: Text(

51 │ AppLocalizations.of(context).mainScreen\_xemtoanbo,

52 │ style: Theme.of(context).textTheme.bodySmall!.copyWith(

53 │ color: AppColors.blueHis2,

54 │ fontStyle: FontStyle.italic,

55 │ ),

56 │ ),

57 │ ),

58 │ ),

59 │ ],

60 │ ),

61 │ ),

62 │

63 │ StreamBuilder<List<AdsModel>>(

64 │ stream: context.read<HomeBloc>().adsStream,

65 │ builder: (context, snapshot) {

66 │ if (snapshot.connectionState == ConnectionState.waiting) {

67 │ return \_buildShimmerLoading();

68 │ } else if (snapshot.hasError) {

69 │ return \_buildErrorState(context, snapshot.error.toString());

70 │ } else if (!snapshot.hasData || snapshot.data!.isEmpty) {

71 │ return \_buildEmptyState(context);

72 │ } else {

73 │ return \_buildAdsList(context, snapshot.data!);

74 │ }

75 │ },

76 │ ),

77 │ //\_buildContent(context, state),

78 │ ],

79 │ );

80 │ },

81 │ );

82 │ }

83 │

84 │ Future<void> \_launchUrlYTB() async {

85 │ final Uri url = Uri.parse(

86 │ 'https://youtube.com/@honghunghospital7846?si=LpZSsyZML9JmjsbB');

87 │

88 │ // Sử dụng LaunchMode.externalNonBrowserApplication hoặc LaunchMode.externalApplication

89 │ if (!await launchUrl(

18751 │ String? upper, {

18752 │ bool includeLower = true,

18753 │ bool includeUpper = true,

18754 │ bool caseSensitive = true,

18755 │ }) {

18756 │ return QueryBuilder.apply(this, (query) {

18757 │ return query.addFilterCondition(FilterCondition.between(

18758 │ property: r'ngaychidinh',

18759 │ lower: lower,

18760 │ includeLower: includeLower,

18761 │ upper: upper,

18762 │ includeUpper: includeUpper,

18763 │ caseSensitive: caseSensitive,

18764 │ ));

18765 │ });

18766 │ }

18767 │

18768 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QAfterFilterCondition>

18769 │ ngaychidinhStartsWith(

18770 │ String value, {

18771 │ bool caseSensitive = true,

18772 │ }) {

18773 │ return QueryBuilder.apply(this, (query) {

18774 │ return query.addFilterCondition(FilterCondition.startsWith(

18775 │ property: r'ngaychidinh',

18776 │ value: value,

18777 │ caseSensitive: caseSensitive,

18778 │ ));

18779 │ });

18780 │ }

18781 │

18782 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QAfterFilterCondition>

18783 │ ngaychidinhEndsWith(

18784 │ String value, {

18785 │ bool caseSensitive = true,

18786 │ }) {

18787 │ return QueryBuilder.apply(this, (query) {

18788 │ return query.addFilterCondition(FilterCondition.endsWith(

18789 │ property: r'ngaychidinh',

18790 │ value: value,

18791 │ caseSensitive: caseSensitive,

18792 │ ));

18793 │ });

18794 │ }

18795 │

18796 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QAfterFilterCondition>

18797 │ ngaychidinhContains(String value, {bool caseSensitive = true}) {

18798 │ return QueryBuilder.apply(this, (query) {

18799 │ return query.addFilterCondition(FilterCondition.contains(

18800 │ property: r'ngaychidinh',

18801 │ value: value,

18802 │ caseSensitive: caseSensitive,

18803 │ ));

18804 │ });

18805 │ }

18806 │

18807 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QAfterFilterCondition>

18808 │ ngaychidinhMatches(String pattern, {bool caseSensitive = true}) {

18809 │ return QueryBuilder.apply(this, (query) {

18810 │ return query.addFilterCondition(FilterCondition.matches(

18811 │ property: r'ngaychidinh',

18812 │ wildcard: pattern,

18813 │ caseSensitive: caseSensitive,

18814 │ ));

18815 │ });

18816 │ }

18817 │

18818 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QAfterFilterCondition>

18819 │ ngaychidinhIsEmpty() {

18820 │ return QueryBuilder.apply(this, (query) {

18821 │ return query.addFilterCondition(FilterCondition.equalTo(

18822 │ property: r'ngaychidinh',

18823 │ value: '',

18824 │ ));

18825 │ });

18826 │ }

18827 │

18828 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QAfterFilterCondition>

18829 │ ngaychidinhIsNotEmpty() {

18830 │ return QueryBuilder.apply(this, (query) {

18831 │ return query.addFilterCondition(FilterCondition.greaterThan(

18832 │ property: r'ngaychidinh',

18833 │ value: '',

18834 │ ));

18835 │ });

18836 │ }

18837 │

18838 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QAfterFilterCondition>

18839 │ ngaygiotraketquaIsNull() {

18840 │ return QueryBuilder.apply(this, (query) {

18841 │ return query.addFilterCondition(const FilterCondition.isNull(

18842 │ property: r'ngaygiotraketqua',

18843 │ ));

18844 │ });

18845 │ }

18846 │

18847 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QAfterFilterCondition>

18848 │ ngaygiotraketquaIsNotNull() {

18849 │ return QueryBuilder.apply(this, (query) {

18850 │ return query.addFilterCondition(const FilterCondition.isNotNull(

18851 │ property: r'ngaygiotraketqua',

18852 │ ));

18853 │ });

18854 │ }

18855 │

18856 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QAfterFilterCondition>

18857 │ ngaygiotraketquaEqualTo(

18858 │ String? value, {

18859 │ bool caseSensitive = true,

18860 │ }) {

18861 │ return QueryBuilder.apply(this, (query) {

18862 │ return query.addFilterCondition(FilterCondition.equalTo(

18863 │ property: r'ngaygiotraketqua',

18864 │ value: value,

18865 │ caseSensitive: caseSensitive,

18866 │ ));

18867 │ });

18868 │ }

18869 │

18870 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QAfterFilterCondition>

18871 │ ngaygiotraketquaGreaterThan(

18872 │ String? value, {

18873 │ bool include = false,

18874 │ bool caseSensitive = true,

18875 │ }) {

18876 │ return QueryBuilder.apply(this, (query) {

18877 │ return query.addFilterCondition(FilterCondition.greaterThan(

18878 │ include: include,

18879 │ property: r'ngaygiotraketqua',

18880 │ value: value,

18881 │ caseSensitive: caseSensitive,

18882 │ ));

18883 │ });

18884 │ }

18885 │

18886 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QAfterFilterCondition>

18887 │ ngaygiotraketquaLessThan(

18888 │ String? value, {

18889 │ bool include = false,

18890 │ bool caseSensitive = true,

18891 │ }) {

18892 │ return QueryBuilder.apply(this, (query) {

18893 │ return query.addFilterCondition(FilterCondition.lessThan(

18894 │ include: include,

18895 │ property: r'ngaygiotraketqua',

18896 │ value: value,

18897 │ caseSensitive: caseSensitive,

18898 │ ));

18899 │ });

18900 │ }

18901 │

18902 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QAfterFilterCondition>

18903 │ ngaygiotraketquaBetween(

18904 │ String? lower,

18905 │ String? upper, {

18906 │ bool includeLower = true,

18907 │ bool includeUpper = true,

18908 │ bool caseSensitive = true,

18909 │ }) {

18910 │ return QueryBuilder.apply(this, (query) {

18911 │ return query.addFilterCondition(FilterCondition.between(

18912 │ property: r'ngaygiotraketqua',

18913 │ lower: lower,

18914 │ includeLower: includeLower,

18915 │ upper: upper,

18916 │ includeUpper: includeUpper,

18917 │ caseSensitive: caseSensitive,

18918 │ ));

18919 │ });

18920 │ }

18921 │

18922 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QAfterFilterCondition>

18923 │ ngaygiotraketquaStartsWith(

18924 │ String value, {

18925 │ bool caseSensitive = true,

18926 │ }) {

18927 │ return QueryBuilder.apply(this, (query) {

18928 │ return query.addFilterCondition(FilterCondition.startsWith(

18929 │ property: r'ngaygiotraketqua',

18930 │ value: value,

18931 │ caseSensitive: caseSensitive,

18932 │ ));

18933 │ });

18934 │ }

18935 │

18936 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QAfterFilterCondition>

18937 │ ngaygiotraketquaEndsWith(

18938 │ String value, {

18939 │ bool caseSensitive = true,

18940 │ }) {

18941 │ return QueryBuilder.apply(this, (query) {

18942 │ return query.addFilterCondition(FilterCondition.endsWith(

18943 │ property: r'ngaygiotraketqua',

18944 │ value: value,

18945 │ caseSensitive: caseSensitive,

18946 │ ));

18947 │ });

18948 │ }

18949 │

18950 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QAfterFilterCondition>

18951 │ ngaygiotraketquaContains(String value, {bool caseSensitive = true}) {

18952 │ return QueryBuilder.apply(this, (query) {

18953 │ return query.addFilterCondition(FilterCondition.contains(

18954 │ property: r'ngaygiotraketqua',

18955 │ value: value,

18956 │ caseSensitive: caseSensitive,

18957 │ ));

18958 │ });

18959 │ }

18960 │

18961 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QAfterFilterCondition>

18962 │ ngaygiotraketquaMatches(String pattern, {bool caseSensitive = true}) {

18963 │ return QueryBuilder.apply(this, (query) {

18964 │ return query.addFilterCondition(FilterCondition.matches(

18965 │ property: r'ngaygiotraketqua',

18966 │ wildcard: pattern,

18967 │ caseSensitive: caseSensitive,

18968 │ ));

18969 │ });

18970 │ }

18971 │

18972 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QAfterFilterCondition>

18973 │ ngaygiotraketquaIsEmpty() {

18974 │ return QueryBuilder.apply(this, (query) {

18975 │ return query.addFilterCondition(FilterCondition.equalTo(

18976 │ property: r'ngaygiotraketqua',

18977 │ value: '',

18978 │ ));

18979 │ });

18980 │ }

18981 │

18982 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QAfterFilterCondition>

18983 │ ngaygiotraketquaIsNotEmpty() {

18984 │ return QueryBuilder.apply(this, (query) {

18985 │ return query.addFilterCondition(FilterCondition.greaterThan(

18986 │ property: r'ngaygiotraketqua',

18987 │ value: '',

18988 │ ));

18989 │ });

18990 │ }

18991 │

18992 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QAfterFilterCondition>

18993 │ ngaythuchienIsNull() {

18994 │ return QueryBuilder.apply(this, (query) {

18995 │ return query.addFilterCondition(const FilterCondition.isNull(

18996 │ property: r'ngaythuchien',

18997 │ ));

18998 │ });

18999 │ }

19000 │

19001 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QAfterFilterCondition>

19002 │ ngaythuchienIsNotNull() {

19003 │ return QueryBuilder.apply(this, (query) {

19004 │ return query.addFilterCondition(const FilterCondition.isNotNull(

19005 │ property: r'ngaythuchien',

19006 │ ));

19007 │ });

19008 │ }

19009 │

19010 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QAfterFilterCondition>

19011 │ ngaythuchienEqualTo(

19012 │ String? value, {

19013 │ bool caseSensitive = true,

19014 │ }) {

19015 │ return QueryBuilder.apply(this, (query) {

19016 │ return query.addFilterCondition(FilterCondition.equalTo(

19017 │ property: r'ngaythuchien',

19018 │ value: value,

19019 │ caseSensitive: caseSensitive,

19020 │ ));

19021 │ });

19022 │ }

19023 │

19024 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QAfterFilterCondition>

19025 │ ngaythuchienGreaterThan(

19026 │ String? value, {

19027 │ bool include = false,

19028 │ bool caseSensitive = true,

19029 │ }) {

19030 │ return QueryBuilder.apply(this, (query) {

19031 │ return query.addFilterCondition(FilterCondition.greaterThan(

19032 │ include: include,

19033 │ property: r'ngaythuchien',

19034 │ value: value,

19035 │ caseSensitive: caseSensitive,

19036 │ ));

19037 │ });

19038 │ }

19039 │

19040 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QAfterFilterCondition>

19041 │ ngaythuchienLessThan(

19042 │ String? value, {

19043 │ bool include = false,

19044 │ bool caseSensitive = true,

19045 │ }) {

19046 │ return QueryBuilder.apply(this, (query) {

19047 │ return query.addFilterCondition(FilterCondition.lessThan(

19048 │ include: include,

19049 │ property: r'ngaythuchien',

19050 │ value: value,

19051 │ caseSensitive: caseSensitive,

19052 │ ));

19053 │ });

19054 │ }

19055 │

19056 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QAfterFilterCondition>

19057 │ ngaythuchienBetween(

19058 │ String? lower,

19059 │ String? upper, {

19060 │ bool includeLower = true,

19061 │ bool includeUpper = true,

19062 │ bool caseSensitive = true,

19063 │ }) {

19064 │ return QueryBuilder.apply(this, (query) {

19065 │ return query.addFilterCondition(FilterCondition.between(

19066 │ property: r'ngaythuchien',

19067 │ lower: lower,

19068 │ includeLower: includeLower,

19069 │ upper: upper,

19070 │ includeUpper: includeUpper,

19071 │ caseSensitive: caseSensitive,

19072 │ ));

19073 │ });

19074 │ }

19075 │

19076 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QAfterFilterCondition>

19077 │ ngaythuchienStartsWith(

19078 │ String value, {

19079 │ bool caseSensitive = true,

19080 │ }) {

19081 │ return QueryBuilder.apply(this, (query) {

19082 │ return query.addFilterCondition(FilterCondition.startsWith(

19083 │ property: r'ngaythuchien',

19084 │ value: value,

19085 │ caseSensitive: caseSensitive,

19086 │ ));

19087 │ });

19088 │ }

19089 │

19090 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QAfterFilterCondition>

19091 │ ngaythuchienEndsWith(

19092 │ String value, {

19093 │ bool caseSensitive = true,

19094 │ }) {

19095 │ return QueryBuilder.apply(this, (query) {

19096 │ return query.addFilterCondition(FilterCondition.endsWith(

19097 │ property: r'ngaythuchien',

19098 │ value: value,

19099 │ caseSensitive: caseSensitive,

19100 │ ));

19101 │ });

19102 │ }

19103 │

19104 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QAfterFilterCondition>

19105 │ ngaythuchienContains(String value, {bool caseSensitive = true}) {

19106 │ return QueryBuilder.apply(this, (query) {

19107 │ return query.addFilterCondition(FilterCondition.contains(

19108 │ property: r'ngaythuchien',

19109 │ value: value,

19110 │ caseSensitive: caseSensitive,

19111 │ ));

19112 │ });

19113 │ }

19114 │

19115 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QAfterFilterCondition>

19116 │ ngaythuchienMatches(String pattern, {bool caseSensitive = true}) {

19117 │ return QueryBuilder.apply(this, (query) {

19118 │ return query.addFilterCondition(FilterCondition.matches(

19119 │ property: r'ngaythuchien',

19120 │ wildcard: pattern,

19121 │ caseSensitive: caseSensitive,

19122 │ ));

19123 │ });

19124 │ }

19125 │

19126 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QAfterFilterCondition>

19127 │ ngaythuchienIsEmpty() {

19128 │ return QueryBuilder.apply(this, (query) {

19129 │ return query.addFilterCondition(FilterCondition.equalTo(

19130 │ property: r'ngaythuchien',

19131 │ value: '',

19132 │ ));

19133 │ });

19134 │ }

19135 │

19136 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QAfterFilterCondition>

19137 │ ngaythuchienIsNotEmpty() {

19138 │ return QueryBuilder.apply(this, (query) {

19139 │ return query.addFilterCondition(FilterCondition.greaterThan(

19140 │ property: r'ngaythuchien',

19141 │ value: '',

19142 │ ));

19143 │ });

19144 │ }

19145 │

19146 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QAfterFilterCondition>

19147 │ ngaytruongkhoakyIsNull() {

19148 │ return QueryBuilder.apply(this, (query) {

19149 │ return query.addFilterCondition(const FilterCondition.isNull(

19150 │ property: r'ngaytruongkhoaky',

19151 │ ));

19152 │ });

19153 │ }

19154 │

19155 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QAfterFilterCondition>

19156 │ ngaytruongkhoakyIsNotNull() {

19157 │ return QueryBuilder.apply(this, (query) {

19158 │ return query.addFilterCondition(const FilterCondition.isNotNull(

19159 │ property: r'ngaytruongkhoaky',

19160 │ ));

19161 │ });

19162 │ }

19163 │

19164 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QAfterFilterCondition>

19165 │ ngaytruongkhoakyEqualTo(

19166 │ String? value, {

19167 │ bool caseSensitive = true,

19168 │ }) {

19169 │ return QueryBuilder.apply(this, (query) {

19170 │ return query.addFilterCondition(FilterCondition.equalTo(

19171 │ property: r'ngaytruongkhoaky',

19172 │ value: value,

19173 │ caseSensitive: caseSensitive,

19174 │ ));

19175 │ });

19176 │ }

19177 │

19178 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QAfterFilterCondition>

19179 │ ngaytruongkhoakyGreaterThan(

19180 │ String? value, {

19181 │ bool include = false,

19182 │ bool caseSensitive = true,

19183 │ }) {

19184 │ return QueryBuilder.apply(this, (query) {

19185 │ return query.addFilterCondition(FilterCondition.greaterThan(

19186 │ include: include,

19187 │ property: r'ngaytruongkhoaky',

19188 │ value: value,

19189 │ caseSensitive: caseSensitive,

19190 │ ));

19191 │ });

19192 │ }

19193 │

19194 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QAfterFilterCondition>

19195 │ ngaytruongkhoakyLessThan(

19196 │ String? value, {

19197 │ bool include = false,

19198 │ bool caseSensitive = true,

19199 │ }) {

19200 │ return QueryBuilder.apply(this, (query) {

19201 │ return query.addFilterCondition(FilterCondition.lessThan(

19202 │ include: include,

19203 │ property: r'ngaytruongkhoaky',

19204 │ value: value,

19205 │ caseSensitive: caseSensitive,

19206 │ ));

19207 │ });

19208 │ }

19209 │

19210 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QAfterFilterCondition>

19211 │ ngaytruongkhoakyBetween(

19212 │ String? lower,

19213 │ String? upper, {

19214 │ bool includeLower = true,

19215 │ bool includeUpper = true,

19216 │ bool caseSensitive = true,

19217 │ }) {

19218 │ return QueryBuilder.apply(this, (query) {

19219 │ return query.addFilterCondition(FilterCondition.between(

19220 │ property: r'ngaytruongkhoaky',

19221 │ lower: lower,

19222 │ includeLower: includeLower,

19223 │ upper: upper,

19224 │ includeUpper: includeUpper,

19225 │ caseSensitive: caseSensitive,

19226 │ ));

19227 │ });

19228 │ }

19229 │

19230 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QAfterFilterCondition>

19231 │ ngaytruongkhoakyStartsWith(

19232 │ String value, {

19233 │ bool caseSensitive = true,

19234 │ }) {

19235 │ return QueryBuilder.apply(this, (query) {

19236 │ return query.addFilterCondition(FilterCondition.startsWith(

19237 │ property: r'ngaytruongkhoaky',

19238 │ value: value,

19239 │ caseSensitive: caseSensitive,

19240 │ ));

19241 │ });

19242 │ }

19243 │

19244 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QAfterFilterCondition>

19245 │ ngaytruongkhoakyEndsWith(

19246 │ String value, {

19247 │ bool caseSensitive = true,

19248 │ }) {

19249 │ return QueryBuilder.apply(this, (query) {

19250 │ return query.addFilterCondition(FilterCondition.endsWith(

19251 │ property: r'ngaytruongkhoaky',

19252 │ value: value,

19253 │ caseSensitive: caseSensitive,

19254 │ ));

19255 │ });

19256 │ }

19257 │

19258 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QAfterFilterCondition>

19259 │ ngaytruongkhoakyContains(String value, {bool caseSensitive = true}) {

19260 │ return QueryBuilder.apply(this, (query) {

19261 │ return query.addFilterCondition(FilterCondition.contains(

19262 │ property: r'ngaytruongkhoaky',

19263 │ value: value,

19264 │ caseSensitive: caseSensitive,

19265 │ ));

19266 │ });

19267 │ }

19268 │

19269 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QAfterFilterCondition>

19270 │ ngaytruongkhoakyMatches(String pattern, {bool caseSensitive = true}) {

19271 │ return QueryBuilder.apply(this, (query) {

19272 │ return query.addFilterCondition(FilterCondition.matches(

19273 │ property: r'ngaytruongkhoaky',

19274 │ wildcard: pattern,

19275 │ caseSensitive: caseSensitive,

19276 │ ));

19277 │ });

19278 │ }

19279 │

19280 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QAfterFilterCondition>

19281 │ ngaytruongkhoakyIsEmpty() {

19282 │ return QueryBuilder.apply(this, (query) {

19283 │ return query.addFilterCondition(FilterCondition.equalTo(

19284 │ property: r'ngaytruongkhoaky',

19285 │ value: '',

19286 │ ));

19287 │ });

19288 │ }

19289 │

19290 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QAfterFilterCondition>

19291 │ ngaytruongkhoakyIsNotEmpty() {

19292 │ return QueryBuilder.apply(this, (query) {

19293 │ return query.addFilterCondition(FilterCondition.greaterThan(

19294 │ property: r'ngaytruongkhoaky',

19295 │ value: '',

19296 │ ));

19297 │ });

19298 │ }

19299 │

19300 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QAfterFilterCondition>

19301 │ tendichvuIsNull() {

19302 │ return QueryBuilder.apply(this, (query) {

19303 │ return query.addFilterCondition(const FilterCondition.isNull(

19304 │ property: r'tendichvu',

19305 │ ));

19306 │ });

19307 │ }

19308 │

19309 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QAfterFilterCondition>

19310 │ tendichvuIsNotNull() {

19311 │ return QueryBuilder.apply(this, (query) {

19312 │ return query.addFilterCondition(const FilterCondition.isNotNull(

19313 │ property: r'tendichvu',

19314 │ ));

19315 │ });

19316 │ }

19317 │

19318 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QAfterFilterCondition>

19319 │ tendichvuEqualTo(

19320 │ String? value, {

19321 │ bool caseSensitive = true,

19322 │ }) {

19323 │ return QueryBuilder.apply(this, (query) {

19324 │ return query.addFilterCondition(FilterCondition.equalTo(

19325 │ property: r'tendichvu',

19326 │ value: value,

19327 │ caseSensitive: caseSensitive,

19328 │ ));

19329 │ });

19330 │ }

19331 │

19332 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QAfterFilterCondition>

19333 │ tendichvuGreaterThan(

19334 │ String? value, {

19335 │ bool include = false,

19336 │ bool caseSensitive = true,

19337 │ }) {

19338 │ return QueryBuilder.apply(this, (query) {

19339 │ return query.addFilterCondition(FilterCondition.greaterThan(

19340 │ include: include,

19341 │ property: r'tendichvu',

19342 │ value: value,

19343 │ caseSensitive: caseSensitive,

19344 │ ));

19345 │ });

19346 │ }

19347 │

19348 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QAfterFilterCondition>

19349 │ tendichvuLessThan(

19350 │ String? value, {

19351 │ bool include = false,

19352 │ bool caseSensitive = true,

19353 │ }) {

19354 │ return QueryBuilder.apply(this, (query) {

19355 │ return query.addFilterCondition(FilterCondition.lessThan(

19356 │ include: include,

19357 │ property: r'tendichvu',

19358 │ value: value,

19359 │ caseSensitive: caseSensitive,

19360 │ ));

19361 │ });

19362 │ }

19363 │

19364 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QAfterFilterCondition>

19365 │ tendichvuBetween(

19366 │ String? lower,

19367 │ String? upper, {

19368 │ bool includeLower = true,

19369 │ bool includeUpper = true,

19370 │ bool caseSensitive = true,

19371 │ }) {

19372 │ return QueryBuilder.apply(this, (query) {

19373 │ return query.addFilterCondition(FilterCondition.between(

19374 │ property: r'tendichvu',

19375 │ lower: lower,

19376 │ includeLower: includeLower,

19377 │ upper: upper,

19378 │ includeUpper: includeUpper,

19379 │ caseSensitive: caseSensitive,

19380 │ ));

19381 │ });

19382 │ }

19383 │

19384 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QAfterFilterCondition>

19385 │ tendichvuStartsWith(

19386 │ String value, {

19387 │ bool caseSensitive = true,

19388 │ }) {

19389 │ return QueryBuilder.apply(this, (query) {

19390 │ return query.addFilterCondition(FilterCondition.startsWith(

19391 │ property: r'tendichvu',

19392 │ value: value,

19393 │ caseSensitive: caseSensitive,

19394 │ ));

19395 │ });

19396 │ }

19397 │

19398 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QAfterFilterCondition>

19399 │ tendichvuEndsWith(

19400 │ String value, {

19401 │ bool caseSensitive = true,

19402 │ }) {

19403 │ return QueryBuilder.apply(this, (query) {

19404 │ return query.addFilterCondition(FilterCondition.endsWith(

19405 │ property: r'tendichvu',

19406 │ value: value,

19407 │ caseSensitive: caseSensitive,

19408 │ ));

19409 │ });

19410 │ }

19411 │

19412 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QAfterFilterCondition>

19413 │ tendichvuContains(String value, {bool caseSensitive = true}) {

19414 │ return QueryBuilder.apply(this, (query) {

19415 │ return query.addFilterCondition(FilterCondition.contains(

19416 │ property: r'tendichvu',

19417 │ value: value,

19418 │ caseSensitive: caseSensitive,

19419 │ ));

19420 │ });

19421 │ }

19422 │

19423 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QAfterFilterCondition>

19424 │ tendichvuMatches(String pattern, {bool caseSensitive = true}) {

19425 │ return QueryBuilder.apply(this, (query) {

19426 │ return query.addFilterCondition(FilterCondition.matches(

19427 │ property: r'tendichvu',

19428 │ wildcard: pattern,

19429 │ caseSensitive: caseSensitive,

19430 │ ));

19431 │ });

19432 │ }

19433 │

19434 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QAfterFilterCondition>

19435 │ tendichvuIsEmpty() {

19436 │ return QueryBuilder.apply(this, (query) {

19437 │ return query.addFilterCondition(FilterCondition.equalTo(

19438 │ property: r'tendichvu',

19439 │ value: '',

19440 │ ));

19441 │ });

19442 │ }

19443 │

19444 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QAfterFilterCondition>

19445 │ tendichvuIsNotEmpty() {

19446 │ return QueryBuilder.apply(this, (query) {

19447 │ return query.addFilterCondition(FilterCondition.greaterThan(

19448 │ property: r'tendichvu',

19449 │ value: '',

19450 │ ));

19451 │ });

19452 │ }

19453 │ }

19454 │

19455 │ extension CDHAResultXNIsarQueryObject

19456 │ on QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QFilterCondition> {}

19457 │

19458 │ extension CDHAResultXNIsarQueryLinks

19459 │ on QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QFilterCondition> {

19460 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QAfterFilterCondition>

19461 │ laboratoryResult(FilterQuery<LaboratoryResultIsar> q) {

19462 │ return QueryBuilder.apply(this, (query) {

19463 │ return query.link(q, r'laboratoryResult');

19464 │ });

19465 │ }

19466 │

19467 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QAfterFilterCondition>

19468 │ laboratoryResultIsNull() {

19469 │ return QueryBuilder.apply(this, (query) {

19470 │ return query.linkLength(r'laboratoryResult', 0, true, 0, true);

19471 │ });

19472 │ }

19473 │ }

19474 │

19475 │ extension CDHAResultXNIsarQuerySortBy

19476 │ on QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QSortBy> {

19477 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QAfterSortBy>

19478 │ sortByBacsithuchien() {

19479 │ return QueryBuilder.apply(this, (query) {

19480 │ return query.addSortBy(r'bacsithuchien', Sort.asc);

19481 │ });

19482 │ }

19483 │

19484 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QAfterSortBy>

19485 │ sortByBacsithuchienDesc() {

19486 │ return QueryBuilder.apply(this, (query) {

19487 │ return query.addSortBy(r'bacsithuchien', Sort.desc);

19488 │ });

19489 │ }

19490 │

19491 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QAfterSortBy>

19492 │ sortByChukybacsi() {

19493 │ return QueryBuilder.apply(this, (query) {

19494 │ return query.addSortBy(r'chukybacsi', Sort.asc);

19495 │ });

19496 │ }

19497 │

19498 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QAfterSortBy>

19499 │ sortByChukybacsiDesc() {

19500 │ return QueryBuilder.apply(this, (query) {

19501 │ return query.addSortBy(r'chukybacsi', Sort.desc);

19502 │ });

19503 │ }

19504 │

19505 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QAfterSortBy>

19506 │ sortByChukyktv() {

19507 │ return QueryBuilder.apply(this, (query) {

19508 │ return query.addSortBy(r'chukyktv', Sort.asc);

19509 │ });

19510 │ }

19511 │

19512 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QAfterSortBy>

19513 │ sortByChukyktvDesc() {

19514 │ return QueryBuilder.apply(this, (query) {

19515 │ return query.addSortBy(r'chukyktv', Sort.desc);

19516 │ });

19517 │ }

19518 │

19519 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QAfterSortBy>

19520 │ sortByFileketqua() {

19521 │ return QueryBuilder.apply(this, (query) {

19522 │ return query.addSortBy(r'fileketqua', Sort.asc);

19523 │ });

19524 │ }

19525 │

19526 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QAfterSortBy>

19527 │ sortByFileketquaDesc() {

19528 │ return QueryBuilder.apply(this, (query) {

19529 │ return query.addSortBy(r'fileketqua', Sort.desc);

19530 │ });

19531 │ }

19532 │

19533 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QAfterSortBy>

19534 │ sortByFilepath() {

19535 │ return QueryBuilder.apply(this, (query) {

19536 │ return query.addSortBy(r'filepath', Sort.asc);

19537 │ });

19538 │ }

19539 │

19540 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QAfterSortBy>

19541 │ sortByFilepathDesc() {

19542 │ return QueryBuilder.apply(this, (query) {

19543 │ return query.addSortBy(r'filepath', Sort.desc);

19544 │ });

19545 │ }

19546 │

19547 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QAfterSortBy>

19548 │ sortByHinhanh() {

19549 │ return QueryBuilder.apply(this, (query) {

19550 │ return query.addSortBy(r'hinhanh', Sort.asc);

19551 │ });

19552 │ }

19553 │

19554 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QAfterSortBy>

19555 │ sortByHinhanhDesc() {

19556 │ return QueryBuilder.apply(this, (query) {

19557 │ return query.addSortBy(r'hinhanh', Sort.desc);

19558 │ });

19559 │ }

19560 │

19561 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QAfterSortBy>

19562 │ sortByHinhanhinkq() {

19563 │ return QueryBuilder.apply(this, (query) {

19564 │ return query.addSortBy(r'hinhanhinkq', Sort.asc);

19565 │ });

19566 │ }

19567 │

19568 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QAfterSortBy>

19569 │ sortByHinhanhinkqDesc() {

19570 │ return QueryBuilder.apply(this, (query) {

19571 │ return query.addSortBy(r'hinhanhinkq', Sort.desc);

19572 │ });

19573 │ }

19574 │

19575 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QAfterSortBy>

19576 │ sortByKetluan() {

19577 │ return QueryBuilder.apply(this, (query) {

19578 │ return query.addSortBy(r'ketluan', Sort.asc);

19579 │ });

19580 │ }

19581 │

19582 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QAfterSortBy>

19583 │ sortByKetluanDesc() {

19584 │ return QueryBuilder.apply(this, (query) {

19585 │ return query.addSortBy(r'ketluan', Sort.desc);

19586 │ });

19587 │ }

19588 │

19589 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QAfterSortBy>

19590 │ sortByKetqua() {

19591 │ return QueryBuilder.apply(this, (query) {

19592 │ return query.addSortBy(r'ketqua', Sort.asc);

19593 │ });

19594 │ }

19595 │

19596 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QAfterSortBy>

19597 │ sortByKetquaDesc() {

19598 │ return QueryBuilder.apply(this, (query) {

19599 │ return query.addSortBy(r'ketqua', Sort.desc);

19600 │ });

19601 │ }

19602 │

19603 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QAfterSortBy>

19604 │ sortByKtvthuchien() {

19605 │ return QueryBuilder.apply(this, (query) {

19606 │ return query.addSortBy(r'ktvthuchien', Sort.asc);

19607 │ });

19608 │ }

19609 │

19610 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QAfterSortBy>

19611 │ sortByKtvthuchienDesc() {

19612 │ return QueryBuilder.apply(this, (query) {

19613 │ return query.addSortBy(r'ktvthuchien', Sort.desc);

19614 │ });

19615 │ }

19616 │

19617 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QAfterSortBy>

19618 │ sortByMaTiepNhan() {

19619 │ return QueryBuilder.apply(this, (query) {

19620 │ return query.addSortBy(r'maTiepNhan', Sort.asc);

19621 │ });

19622 │ }

19623 │

19624 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QAfterSortBy>

19625 │ sortByMaTiepNhanDesc() {

19626 │ return QueryBuilder.apply(this, (query) {

19627 │ return query.addSortBy(r'maTiepNhan', Sort.desc);

19628 │ });

19629 │ }

19630 │

19631 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QAfterSortBy>

19632 │ sortByMachidinh() {

19633 │ return QueryBuilder.apply(this, (query) {

19634 │ return query.addSortBy(r'machidinh', Sort.asc);

19635 │ });

19636 │ }

19637 │

19638 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QAfterSortBy>

19639 │ sortByMachidinhDesc() {

19640 │ return QueryBuilder.apply(this, (query) {

19641 │ return query.addSortBy(r'machidinh', Sort.desc);

19642 │ });

19643 │ }

19644 │

19645 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QAfterSortBy>

19646 │ sortByMadichvu() {

19647 │ return QueryBuilder.apply(this, (query) {

19648 │ return query.addSortBy(r'madichvu', Sort.asc);

19649 │ });

19650 │ }

19651 │

19652 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QAfterSortBy>

19653 │ sortByMadichvuDesc() {

19654 │ return QueryBuilder.apply(this, (query) {

19655 │ return query.addSortBy(r'madichvu', Sort.desc);

19656 │ });

19657 │ }

19658 │

19659 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QAfterSortBy>

19660 │ sortByNgaychidinh() {

19661 │ return QueryBuilder.apply(this, (query) {

19662 │ return query.addSortBy(r'ngaychidinh', Sort.asc);

19663 │ });

19664 │ }

19665 │

19666 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QAfterSortBy>

19667 │ sortByNgaychidinhDesc() {

19668 │ return QueryBuilder.apply(this, (query) {

19669 │ return query.addSortBy(r'ngaychidinh', Sort.desc);

19670 │ });

19671 │ }

19672 │

19673 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QAfterSortBy>

19674 │ sortByNgaygiotraketqua() {

19675 │ return QueryBuilder.apply(this, (query) {

19676 │ return query.addSortBy(r'ngaygiotraketqua', Sort.asc);

19677 │ });

19678 │ }

19679 │

19680 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QAfterSortBy>

19681 │ sortByNgaygiotraketquaDesc() {

19682 │ return QueryBuilder.apply(this, (query) {

19683 │ return query.addSortBy(r'ngaygiotraketqua', Sort.desc);

19684 │ });

19685 │ }

19686 │

19687 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QAfterSortBy>

19688 │ sortByNgaythuchien() {

19689 │ return QueryBuilder.apply(this, (query) {

19690 │ return query.addSortBy(r'ngaythuchien', Sort.asc);

19691 │ });

19692 │ }

19693 │

19694 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QAfterSortBy>

19695 │ sortByNgaythuchienDesc() {

19696 │ return QueryBuilder.apply(this, (query) {

19697 │ return query.addSortBy(r'ngaythuchien', Sort.desc);

19698 │ });

19699 │ }

19700 │

19701 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QAfterSortBy>

19702 │ sortByNgaytruongkhoaky() {

19703 │ return QueryBuilder.apply(this, (query) {

19704 │ return query.addSortBy(r'ngaytruongkhoaky', Sort.asc);

19705 │ });

19706 │ }

19707 │

19708 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QAfterSortBy>

19709 │ sortByNgaytruongkhoakyDesc() {

19710 │ return QueryBuilder.apply(this, (query) {

19711 │ return query.addSortBy(r'ngaytruongkhoaky', Sort.desc);

19712 │ });

19713 │ }

19714 │

19715 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QAfterSortBy>

19716 │ sortByTendichvu() {

19717 │ return QueryBuilder.apply(this, (query) {

19718 │ return query.addSortBy(r'tendichvu', Sort.asc);

19719 │ });

19720 │ }

19721 │

19722 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QAfterSortBy>

19723 │ sortByTendichvuDesc() {

19724 │ return QueryBuilder.apply(this, (query) {

19725 │ return query.addSortBy(r'tendichvu', Sort.desc);

19726 │ });

19727 │ }

19728 │ }

19729 │

19730 │ extension CDHAResultXNIsarQuerySortThenBy

19731 │ on QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QSortThenBy> {

19732 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QAfterSortBy>

19733 │ thenByBacsithuchien() {

19734 │ return QueryBuilder.apply(this, (query) {

19735 │ return query.addSortBy(r'bacsithuchien', Sort.asc);

19736 │ });

19737 │ }

19738 │

19739 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QAfterSortBy>

19740 │ thenByBacsithuchienDesc() {

19741 │ return QueryBuilder.apply(this, (query) {

19742 │ return query.addSortBy(r'bacsithuchien', Sort.desc);

19743 │ });

19744 │ }

19745 │

19746 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QAfterSortBy>

19747 │ thenByChukybacsi() {

19748 │ return QueryBuilder.apply(this, (query) {

19749 │ return query.addSortBy(r'chukybacsi', Sort.asc);

19750 │ });

19751 │ }

19752 │

19753 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QAfterSortBy>

19754 │ thenByChukybacsiDesc() {

19755 │ return QueryBuilder.apply(this, (query) {

19756 │ return query.addSortBy(r'chukybacsi', Sort.desc);

19757 │ });

19758 │ }

19759 │

19760 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QAfterSortBy>

19761 │ thenByChukyktv() {

19762 │ return QueryBuilder.apply(this, (query) {

19763 │ return query.addSortBy(r'chukyktv', Sort.asc);

19764 │ });

19765 │ }

19766 │

19767 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QAfterSortBy>

19768 │ thenByChukyktvDesc() {

19769 │ return QueryBuilder.apply(this, (query) {

19770 │ return query.addSortBy(r'chukyktv', Sort.desc);

19771 │ });

19772 │ }

19773 │

19774 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QAfterSortBy>

19775 │ thenByFileketqua() {

19776 │ return QueryBuilder.apply(this, (query) {

19777 │ return query.addSortBy(r'fileketqua', Sort.asc);

19778 │ });

19779 │ }

19780 │

19781 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QAfterSortBy>

19782 │ thenByFileketquaDesc() {

19783 │ return QueryBuilder.apply(this, (query) {

19784 │ return query.addSortBy(r'fileketqua', Sort.desc);

19785 │ });

19786 │ }

19787 │

19788 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QAfterSortBy>

19789 │ thenByFilepath() {

19790 │ return QueryBuilder.apply(this, (query) {

19791 │ return query.addSortBy(r'filepath', Sort.asc);

19792 │ });

19793 │ }

19794 │

19795 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QAfterSortBy>

19796 │ thenByFilepathDesc() {

19797 │ return QueryBuilder.apply(this, (query) {

19798 │ return query.addSortBy(r'filepath', Sort.desc);

19799 │ });

19800 │ }

19801 │

19802 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QAfterSortBy>

19803 │ thenByHinhanh() {

19804 │ return QueryBuilder.apply(this, (query) {

19805 │ return query.addSortBy(r'hinhanh', Sort.asc);

19806 │ });

19807 │ }

19808 │

19809 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QAfterSortBy>

19810 │ thenByHinhanhDesc() {

19811 │ return QueryBuilder.apply(this, (query) {

19812 │ return query.addSortBy(r'hinhanh', Sort.desc);

19813 │ });

19814 │ }

19815 │

19816 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QAfterSortBy>

19817 │ thenByHinhanhinkq() {

19818 │ return QueryBuilder.apply(this, (query) {

19819 │ return query.addSortBy(r'hinhanhinkq', Sort.asc);

19820 │ });

19821 │ }

19822 │

19823 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QAfterSortBy>

19824 │ thenByHinhanhinkqDesc() {

19825 │ return QueryBuilder.apply(this, (query) {

19826 │ return query.addSortBy(r'hinhanhinkq', Sort.desc);

19827 │ });

19828 │ }

19829 │

19830 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QAfterSortBy> thenById() {

19831 │ return QueryBuilder.apply(this, (query) {

19832 │ return query.addSortBy(r'id', Sort.asc);

19833 │ });

19834 │ }

19835 │

19836 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QAfterSortBy>

19837 │ thenByIdDesc() {

19838 │ return QueryBuilder.apply(this, (query) {

19839 │ return query.addSortBy(r'id', Sort.desc);

19840 │ });

19841 │ }

19842 │

19843 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QAfterSortBy>

19844 │ thenByKetluan() {

19845 │ return QueryBuilder.apply(this, (query) {

19846 │ return query.addSortBy(r'ketluan', Sort.asc);

19847 │ });

19848 │ }

19849 │

19850 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QAfterSortBy>

19851 │ thenByKetluanDesc() {

19852 │ return QueryBuilder.apply(this, (query) {

19853 │ return query.addSortBy(r'ketluan', Sort.desc);

19854 │ });

19855 │ }

19856 │

19857 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QAfterSortBy>

19858 │ thenByKetqua() {

19859 │ return QueryBuilder.apply(this, (query) {

19860 │ return query.addSortBy(r'ketqua', Sort.asc);

19861 │ });

19862 │ }

19863 │

19864 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QAfterSortBy>

19865 │ thenByKetquaDesc() {

19866 │ return QueryBuilder.apply(this, (query) {

19867 │ return query.addSortBy(r'ketqua', Sort.desc);

19868 │ });

19869 │ }

19870 │

19871 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QAfterSortBy>

19872 │ thenByKtvthuchien() {

19873 │ return QueryBuilder.apply(this, (query) {

19874 │ return query.addSortBy(r'ktvthuchien', Sort.asc);

19875 │ });

19876 │ }

19877 │

19878 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QAfterSortBy>

19879 │ thenByKtvthuchienDesc() {

19880 │ return QueryBuilder.apply(this, (query) {

19881 │ return query.addSortBy(r'ktvthuchien', Sort.desc);

19882 │ });

19883 │ }

19884 │

19885 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QAfterSortBy>

19886 │ thenByMaTiepNhan() {

19887 │ return QueryBuilder.apply(this, (query) {

19888 │ return query.addSortBy(r'maTiepNhan', Sort.asc);

19889 │ });

19890 │ }

19891 │

19892 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QAfterSortBy>

19893 │ thenByMaTiepNhanDesc() {

19894 │ return QueryBuilder.apply(this, (query) {

19895 │ return query.addSortBy(r'maTiepNhan', Sort.desc);

19896 │ });

19897 │ }

19898 │

19899 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QAfterSortBy>

19900 │ thenByMachidinh() {

19901 │ return QueryBuilder.apply(this, (query) {

19902 │ return query.addSortBy(r'machidinh', Sort.asc);

19903 │ });

19904 │ }

19905 │

19906 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QAfterSortBy>

19907 │ thenByMachidinhDesc() {

19908 │ return QueryBuilder.apply(this, (query) {

19909 │ return query.addSortBy(r'machidinh', Sort.desc);

19910 │ });

19911 │ }

19912 │

19913 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QAfterSortBy>

19914 │ thenByMadichvu() {

19915 │ return QueryBuilder.apply(this, (query) {

19916 │ return query.addSortBy(r'madichvu', Sort.asc);

19917 │ });

19918 │ }

19919 │

19920 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QAfterSortBy>

19921 │ thenByMadichvuDesc() {

19922 │ return QueryBuilder.apply(this, (query) {

19923 │ return query.addSortBy(r'madichvu', Sort.desc);

19924 │ });

19925 │ }

19926 │

19927 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QAfterSortBy>

19928 │ thenByNgaychidinh() {

19929 │ return QueryBuilder.apply(this, (query) {

19930 │ return query.addSortBy(r'ngaychidinh', Sort.asc);

19931 │ });

19932 │ }

19933 │

19934 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QAfterSortBy>

19935 │ thenByNgaychidinhDesc() {

19936 │ return QueryBuilder.apply(this, (query) {

19937 │ return query.addSortBy(r'ngaychidinh', Sort.desc);

19938 │ });

19939 │ }

19940 │

19941 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QAfterSortBy>

19942 │ thenByNgaygiotraketqua() {

19943 │ return QueryBuilder.apply(this, (query) {

19944 │ return query.addSortBy(r'ngaygiotraketqua', Sort.asc);

19945 │ });

19946 │ }

19947 │

19948 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QAfterSortBy>

19949 │ thenByNgaygiotraketquaDesc() {

19950 │ return QueryBuilder.apply(this, (query) {

19951 │ return query.addSortBy(r'ngaygiotraketqua', Sort.desc);

19952 │ });

19953 │ }

19954 │

19955 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QAfterSortBy>

19956 │ thenByNgaythuchien() {

19957 │ return QueryBuilder.apply(this, (query) {

19958 │ return query.addSortBy(r'ngaythuchien', Sort.asc);

19959 │ });

19960 │ }

19961 │

19962 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QAfterSortBy>

19963 │ thenByNgaythuchienDesc() {

19964 │ return QueryBuilder.apply(this, (query) {

19965 │ return query.addSortBy(r'ngaythuchien', Sort.desc);

19966 │ });

19967 │ }

19968 │

19969 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QAfterSortBy>

19970 │ thenByNgaytruongkhoaky() {

19971 │ return QueryBuilder.apply(this, (query) {

19972 │ return query.addSortBy(r'ngaytruongkhoaky', Sort.asc);

19973 │ });

19974 │ }

19975 │

19976 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QAfterSortBy>

19977 │ thenByNgaytruongkhoakyDesc() {

19978 │ return QueryBuilder.apply(this, (query) {

19979 │ return query.addSortBy(r'ngaytruongkhoaky', Sort.desc);

19980 │ });

19981 │ }

19982 │

19983 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QAfterSortBy>

19984 │ thenByTendichvu() {

19985 │ return QueryBuilder.apply(this, (query) {

19986 │ return query.addSortBy(r'tendichvu', Sort.asc);

19987 │ });

19988 │ }

19989 │

19990 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QAfterSortBy>

19991 │ thenByTendichvuDesc() {

19992 │ return QueryBuilder.apply(this, (query) {

19993 │ return query.addSortBy(r'tendichvu', Sort.desc);

19994 │ });

19995 │ }

19996 │ }

19997 │

19998 │ extension CDHAResultXNIsarQueryWhereDistinct

19999 │ on QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QDistinct> {

20000 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QDistinct>

20001 │ distinctByBacsithuchien({bool caseSensitive = true}) {

20002 │ return QueryBuilder.apply(this, (query) {

20003 │ return query.addDistinctBy(r'bacsithuchien',

20004 │ caseSensitive: caseSensitive);

20005 │ });

20006 │ }

20007 │

20008 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QDistinct>

20009 │ distinctByChukybacsi({bool caseSensitive = true}) {

20010 │ return QueryBuilder.apply(this, (query) {

20011 │ return query.addDistinctBy(r'chukybacsi', caseSensitive: caseSensitive);

20012 │ });

20013 │ }

20014 │

20015 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QDistinct>

20016 │ distinctByChukyktv({bool caseSensitive = true}) {

20017 │ return QueryBuilder.apply(this, (query) {

20018 │ return query.addDistinctBy(r'chukyktv', caseSensitive: caseSensitive);

20019 │ });

20020 │ }

20021 │

20022 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QDistinct>

20023 │ distinctByFileketqua({bool caseSensitive = true}) {

20024 │ return QueryBuilder.apply(this, (query) {

20025 │ return query.addDistinctBy(r'fileketqua', caseSensitive: caseSensitive);

20026 │ });

20027 │ }

20028 │

20029 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QDistinct>

20030 │ distinctByFilepath({bool caseSensitive = true}) {

20031 │ return QueryBuilder.apply(this, (query) {

20032 │ return query.addDistinctBy(r'filepath', caseSensitive: caseSensitive);

20033 │ });

20034 │ }

20035 │

20036 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QDistinct> distinctByHinhanh(

20037 │ {bool caseSensitive = true}) {

20038 │ return QueryBuilder.apply(this, (query) {

20039 │ return query.addDistinctBy(r'hinhanh', caseSensitive: caseSensitive);

20040 │ });

20041 │ }

20042 │

20043 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QDistinct>

20044 │ distinctByHinhanhinkq({bool caseSensitive = true}) {

20045 │ return QueryBuilder.apply(this, (query) {

20046 │ return query.addDistinctBy(r'hinhanhinkq', caseSensitive: caseSensitive);

20047 │ });

20048 │ }

20049 │

20050 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QDistinct> distinctByKetluan(

20051 │ {bool caseSensitive = true}) {

20052 │ return QueryBuilder.apply(this, (query) {

20053 │ return query.addDistinctBy(r'ketluan', caseSensitive: caseSensitive);

20054 │ });

20055 │ }

20056 │

20057 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QDistinct> distinctByKetqua(

20058 │ {bool caseSensitive = true}) {

20059 │ return QueryBuilder.apply(this, (query) {

20060 │ return query.addDistinctBy(r'ketqua', caseSensitive: caseSensitive);

20061 │ });

20062 │ }

20063 │

20064 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QDistinct>

20065 │ distinctByKtvthuchien({bool caseSensitive = true}) {

20066 │ return QueryBuilder.apply(this, (query) {

20067 │ return query.addDistinctBy(r'ktvthuchien', caseSensitive: caseSensitive);

20068 │ });

20069 │ }

20070 │

20071 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QDistinct>

20072 │ distinctByMaTiepNhan({bool caseSensitive = true}) {

20073 │ return QueryBuilder.apply(this, (query) {

20074 │ return query.addDistinctBy(r'maTiepNhan', caseSensitive: caseSensitive);

20075 │ });

20076 │ }

20077 │

20078 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QDistinct>

20079 │ distinctByMachidinh({bool caseSensitive = true}) {

20080 │ return QueryBuilder.apply(this, (query) {

20081 │ return query.addDistinctBy(r'machidinh', caseSensitive: caseSensitive);

20082 │ });

20083 │ }

20084 │

20085 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QDistinct>

20086 │ distinctByMadichvu({bool caseSensitive = true}) {

20087 │ return QueryBuilder.apply(this, (query) {

20088 │ return query.addDistinctBy(r'madichvu', caseSensitive: caseSensitive);

20089 │ });

20090 │ }

20091 │

20092 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QDistinct>

20093 │ distinctByNgaychidinh({bool caseSensitive = true}) {

20094 │ return QueryBuilder.apply(this, (query) {

20095 │ return query.addDistinctBy(r'ngaychidinh', caseSensitive: caseSensitive);

20096 │ });

20097 │ }

20098 │

20099 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QDistinct>

20100 │ distinctByNgaygiotraketqua({bool caseSensitive = true}) {

20101 │ return QueryBuilder.apply(this, (query) {

20102 │ return query.addDistinctBy(r'ngaygiotraketqua',

20103 │ caseSensitive: caseSensitive);

20104 │ });

20105 │ }

20106 │

20107 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QDistinct>

20108 │ distinctByNgaythuchien({bool caseSensitive = true}) {

20109 │ return QueryBuilder.apply(this, (query) {

20110 │ return query.addDistinctBy(r'ngaythuchien', caseSensitive: caseSensitive);

20111 │ });

20112 │ }

20113 │

20114 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QDistinct>

20115 │ distinctByNgaytruongkhoaky({bool caseSensitive = true}) {

20116 │ return QueryBuilder.apply(this, (query) {

20117 │ return query.addDistinctBy(r'ngaytruongkhoaky',

20118 │ caseSensitive: caseSensitive);

20119 │ });

20120 │ }

20121 │

20122 │ QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QDistinct>

20123 │ distinctByTendichvu({bool caseSensitive = true}) {

20124 │ return QueryBuilder.apply(this, (query) {

20125 │ return query.addDistinctBy(r'tendichvu', caseSensitive: caseSensitive);

20126 │ });

20127 │ }

20128 │ }

20129 │

20130 │ extension CDHAResultXNIsarQueryProperty

20131 │ on QueryBuilder<CDHAResultXNIsar, CDHAResultXNIsar, QQueryProperty> {

20132 │ QueryBuilder<CDHAResultXNIsar, int, QQueryOperations> idProperty() {

20133 │ return QueryBuilder.apply(this, (query) {

20134 │ return query.addPropertyName(r'id');

20135 │ });

20136 │ }

20137 │

20138 │ QueryBuilder<CDHAResultXNIsar, String?, QQueryOperations>

20139 │ bacsithuchienProperty() {

20140 │ return QueryBuilder.apply(this, (query) {

20141 │ return query.addPropertyName(r'bacsithuchien');

20142 │ });

20143 │ }

20144 │

20145 │ QueryBuilder<CDHAResultXNIsar, String?, QQueryOperations>

20146 │ chukybacsiProperty() {

20147 │ return QueryBuilder.apply(this, (query) {

20148 │ return query.addPropertyName(r'chukybacsi');

20149 │ });

20150 │ }

20151 │

20152 │ QueryBuilder<CDHAResultXNIsar, String?, QQueryOperations> chukyktvProperty() {

20153 │ return QueryBuilder.apply(this, (query) {

20154 │ return query.addPropertyName(r'chukyktv');

20155 │ });

20156 │ }

20157 │

20158 │ QueryBuilder<CDHAResultXNIsar, String?, QQueryOperations>

20159 │ fileketquaProperty() {

20160 │ return QueryBuilder.apply(this, (query) {

20161 │ return query.addPropertyName(r'fileketqua');

20162 │ });

20163 │ }

20164 │

20165 │ QueryBuilder<CDHAResultXNIsar, String?, QQueryOperations> filepathProperty() {

20166 │ return QueryBuilder.apply(this, (query) {

20167 │ return query.addPropertyName(r'filepath');

20168 │ });

20169 │ }

20170 │

20171 │ QueryBuilder<CDHAResultXNIsar, String?, QQueryOperations> hinhanhProperty() {

20172 │ return QueryBuilder.apply(this, (query) {

20173 │ return query.addPropertyName(r'hinhanh');

20174 │ });

20175 │ }

20176 │

20177 │ QueryBuilder<CDHAResultXNIsar, String?, QQueryOperations>

20178 │ hinhanhinkqProperty() {

20179 │ return QueryBuilder.apply(this, (query) {

20180 │ return query.addPropertyName(r'hinhanhinkq');

20181 │ });

20182 │ }

20183 │

20184 │ QueryBuilder<CDHAResultXNIsar, String?, QQueryOperations> ketluanProperty() {

20185 │ return QueryBuilder.apply(this, (query) {

20186 │ return query.addPropertyName(r'ketluan');

20187 │ });

20188 │ }

20189 │

20190 │ QueryBuilder<CDHAResultXNIsar, String?, QQueryOperations> ketquaProperty() {

20191 │ return QueryBuilder.apply(this, (query) {

20192 │ return query.addPropertyName(r'ketqua');

20193 │ });

20194 │ }

20195 │

20196 │ QueryBuilder<CDHAResultXNIsar, String?, QQueryOperations>

20197 │ ktvthuchienProperty() {

20198 │ return QueryBuilder.apply(this, (query) {

20199 │ return query.addPropertyName(r'ktvthuchien');

20200 │ });

20201 │ }

20202 │

20203 │ QueryBuilder<CDHAResultXNIsar, String?, QQueryOperations>

20204 │ maTiepNhanProperty() {

20205 │ return QueryBuilder.apply(this, (query) {

20206 │ return query.addPropertyName(r'maTiepNhan');

20207 │ });

20208 │ }

20209 │

20210 │ QueryBuilder<CDHAResultXNIsar, String?, QQueryOperations>

20211 │ machidinhProperty() {

20212 │ return QueryBuilder.apply(this, (query) {

20213 │ return query.addPropertyName(r'machidinh');

20214 │ });

20215 │ }

20216 │

20217 │ QueryBuilder<CDHAResultXNIsar, String?, QQueryOperations> madichvuProperty() {

20218 │ return QueryBuilder.apply(this, (query) {

20219 │ return query.addPropertyName(r'madichvu');

20220 │ });

20221 │ }

20222 │

20223 │ QueryBuilder<CDHAResultXNIsar, String?, QQueryOperations>

20224 │ ngaychidinhProperty() {

20225 │ return QueryBuilder.apply(this, (query) {

20226 │ return query.addPropertyName(r'ngaychidinh');

20227 │ });

20228 │ }

20229 │

20230 │ QueryBuilder<CDHAResultXNIsar, String?, QQueryOperations>

20231 │ ngaygiotraketquaProperty() {

20232 │ return QueryBuilder.apply(this, (query) {

20233 │ return query.addPropertyName(r'ngaygiotraketqua');

20234 │ });

20235 │ }

20236 │

20237 │ QueryBuilder<CDHAResultXNIsar, String?, QQueryOperations>

20238 │ ngaythuchienProperty() {

20239 │ return QueryBuilder.apply(this, (query) {

20240 │ return query.addPropertyName(r'ngaythuchien');

20241 │ });

20242 │ }

20243 │

20244 │ QueryBuilder<CDHAResultXNIsar, String?, QQueryOperations>

20245 │ ngaytruongkhoakyProperty() {

20246 │ return QueryBuilder.apply(this, (query) {

20247 │ return query.addPropertyName(r'ngaytruongkhoaky');

20248 │ });

20249 │ }

20250 │

20251 │ QueryBuilder<CDHAResultXNIsar, String?, QQueryOperations>

20252 │ tendichvuProperty() {

20253 │ return QueryBuilder.apply(this, (query) {

20254 │ return query.addPropertyName(r'tendichvu');

20255 │ });

20256 │ }

20257 │ }

20258 │

20259 │ // coverage:ignore-file

20260 │ // ignore\_for\_file: duplicate\_ignore, non\_constant\_identifier\_names, constant\_identifier\_names, invalid\_use\_of\_protected...

20261 │

20262 │ extension GetVpntXetNghiemIsarCollection on Isar {

20263 │ IsarCollection<VpntXetNghiemIsar> get vpntXetNghiemIsars => this.collection();

20264 │ }

20265 │

20266 │ const VpntXetNghiemIsarSchema = CollectionSchema(

20267 │ name: r'VpntXetNghiemIsar',

20268 │ id: 8688675209068507424,

20269 │ properties: {

20270 │ r'chandoan': PropertySchema(

20271 │ id: 0,

20272 │ name: r'chandoan',

20273 │ type: IsarType.string,

20274 │ ),

20275 │ r'maTiepNhan': PropertySchema(

20276 │ id: 1,

20277 │ name: r'maTiepNhan',

20278 │ type: IsarType.string,

20279 │ ),

20280 │ r'machidinh': PropertySchema(

20281 │ id: 2,

20282 │ name: r'machidinh',

20283 │ type: IsarType.string,

20284 │ ),

20285 │ r'mavp': PropertySchema(

20286 │ id: 3,

20287 │ name: r'mavp',

20288 │ type: IsarType.string,

20289 │ ),

20290 │ r'phongthuchien': PropertySchema(

20291 │ id: 4,

20292 │ name: r'phongthuchien',

20293 │ type: IsarType.string,

20294 │ ),

20295 │ r'tenbs': PropertySchema(

20296 │ id: 5,

20297 │ name: r'tenbs',

20298 │ type: IsarType.string,

20299 │ ),

20300 │ r'tenkhoa': PropertySchema(

20301 │ id: 6,

20302 │ name: r'tenkhoa',

20303 │ type: IsarType.string,

20304 │ ),

20305 │ r'tenphong': PropertySchema(

20306 │ id: 7,

20307 │ name: r'tenphong',

20308 │ type: IsarType.string,

20309 │ )

20310 │ },

20311 │ estimateSize: \_vpntXetNghiemIsarEstimateSize,

20312 │ serialize: \_vpntXetNghiemIsarSerialize,

20313 │ deserialize: \_vpntXetNghiemIsarDeserialize,

20314 │ deserializeProp: \_vpntXetNghiemIsarDeserializeProp,

20315 │ idName: r'id',

20316 │ indexes: {

20317 │ r'maTiepNhan': IndexSchema(

20318 │ id: -7916468118296143105,

20319 │ name: r'maTiepNhan',

20320 │ unique: false,

20321 │ replace: false,

20322 │ properties: [

20323 │ IndexPropertySchema(

20324 │ name: r'maTiepNhan',

20325 │ type: IndexType.hash,

20326 │ caseSensitive: true,

20327 │ )

20328 │ ],

20329 │ )

20330 │ },

20331 │ links: {

20332 │ r'laboratoryResult': LinkSchema(

20333 │ id: 7037784238101384118,

20334 │ name: r'laboratoryResult',

20335 │ target: r'LaboratoryResultIsar',

20336 │ single: true,

20337 │ linkName: r'vpntList',

20338 │ )

20339 │ },

20340 │ embeddedSchemas: {},

20341 │ getId: \_vpntXetNghiemIsarGetId,

20342 │ getLinks: \_vpntXetNghiemIsarGetLinks,

20343 │ attach: \_vpntXetNghiemIsarAttach,

20344 │ version: '3.1.0+1',

20345 │ );

20346 │

20347 │ int \_vpntXetNghiemIsarEstimateSize(

20348 │ VpntXetNghiemIsar object,

20349 │ List<int> offsets,

20350 │ Map<Type, List<int>> allOffsets,

20351 │ ) {

20352 │ var bytesCount = offsets.last;

20353 │ {

20354 │ final value = object.chandoan;

20355 │ if (value != null) {

20356 │ bytesCount += 3 + value.length \* 3;

20357 │ }

20358 │ }

20359 │ {

20360 │ final value = object.maTiepNhan;

20361 │ if (value != null) {

20362 │ bytesCount += 3 + value.length \* 3;

20363 │ }

20364 │ }

20365 │ {

20366 │ final value = object.machidinh;

20367 │ if (value != null) {

20368 │ bytesCount += 3 + value.length \* 3;

20369 │ }

20370 │ }

20371 │ {

20372 │ final value = object.mavp;

20373 │ if (value != null) {

20374 │ bytesCount += 3 + value.length \* 3;

20375 │ }

20376 │ }

20377 │ {

20378 │ final value = object.phongthuchien;

20379 │ if (value != null) {

20380 │ bytesCount += 3 + value.length \* 3;

20381 │ }

20382 │ }

20383 │ {

20384 │ final value = object.tenbs;

20385 │ if (value != null) {

20386 │ bytesCount += 3 + value.length \* 3;

20387 │ }

20388 │ }

20389 │ {

20390 │ final value = object.tenkhoa;

20391 │ if (value != null) {

20392 │ bytesCount += 3 + value.length \* 3;

20393 │ }

20394 │ }

20395 │ {

20396 │ final value = object.tenphong;

20397 │ if (value != null) {

20398 │ bytesCount += 3 + value.length \* 3;

20399 │ }

20400 │ }

20401 │ return bytesCount;

20402 │ }

20403 │

20404 │ void \_vpntXetNghiemIsarSerialize(

20405 │ VpntXetNghiemIsar object,

20406 │ IsarWriter writer,

20407 │ List<int> offsets,

20408 │ Map<Type, List<int>> allOffsets,

20409 │ ) {

20410 │ writer.writeString(offsets[0], object.chandoan);

20411 │ writer.writeString(offsets[1], object.maTiepNhan);

20412 │ writer.writeString(offsets[2], object.machidinh);

20413 │ writer.writeString(offsets[3], object.mavp);

20414 │ writer.writeString(offsets[4], object.phongthuchien);

20415 │ writer.writeString(offsets[5], object.tenbs);

20416 │ writer.writeString(offsets[6], object.tenkhoa);

20417 │ writer.writeString(offsets[7], object.tenphong);

20418 │ }

20419 │

20420 │ VpntXetNghiemIsar \_vpntXetNghiemIsarDeserialize(

20421 │ Id id,

20422 │ IsarReader reader,

20423 │ List<int> offsets,

20424 │ Map<Type, List<int>> allOffsets,

20425 │ ) {

20426 │ final object = VpntXetNghiemIsar();

20427 │ object.chandoan = reader.readStringOrNull(offsets[0]);

20428 │ object.id = id;

20429 │ object.maTiepNhan = reader.readStringOrNull(offsets[1]);

20430 │ object.machidinh = reader.readStringOrNull(offsets[2]);

20431 │ object.mavp = reader.readStringOrNull(offsets[3]);

20432 │ object.phongthuchien = reader.readStringOrNull(offsets[4]);

20433 │ object.tenbs = reader.readStringOrNull(offsets[5]);

20434 │ object.tenkhoa = reader.readStringOrNull(offsets[6]);

20435 │ object.tenphong = reader.readStringOrNull(offsets[7]);

20436 │ return object;

20437 │ }

20438 │

20439 │ P \_vpntXetNghiemIsarDeserializeProp<P>(

20440 │ IsarReader reader,

20441 │ int propertyId,

20442 │ int offset,

20443 │ Map<Type, List<int>> allOffsets,

20444 │ ) {

20445 │ switch (propertyId) {

20446 │ case 0:

20447 │ return (reader.readStringOrNull(offset)) as P;

20448 │ case 1:

20449 │ return (reader.readStringOrNull(offset)) as P;

20450 │ case 2:

20451 │ return (reader.readStringOrNull(offset)) as P;

20452 │ case 3:

20453 │ return (reader.readStringOrNull(offset)) as P;

20454 │ case 4:

20455 │ return (reader.readStringOrNull(offset)) as P;

20456 │ case 5:

20457 │ return (reader.readStringOrNull(offset)) as P;

20458 │ case 6:

20459 │ return (reader.readStringOrNull(offset)) as P;

20460 │ case 7:

20461 │ return (reader.readStringOrNull(offset)) as P;

20462 │ default:

20463 │ throw IsarError('Unknown property with id $propertyId');

20464 │ }

20465 │ }

20466 │

20467 │ Id \_vpntXetNghiemIsarGetId(VpntXetNghiemIsar object) {

20468 │ return object.id;

20469 │ }

20470 │

20471 │ List<IsarLinkBase<dynamic>> \_vpntXetNghiemIsarGetLinks(

20472 │ VpntXetNghiemIsar object) {

20473 │ return [object.laboratoryResult];

20474 │ }

20475 │

20476 │ void \_vpntXetNghiemIsarAttach(

20477 │ IsarCollection<dynamic> col, Id id, VpntXetNghiemIsar object) {

20478 │ object.id = id;

20479 │ object.laboratoryResult.attach(col,

20480 │ col.isar.collection<LaboratoryResultIsar>(), r'laboratoryResult', id);

20481 │ }

20482 │

20483 │ extension VpntXetNghiemIsarQueryWhereSort

20484 │ on QueryBuilder<VpntXetNghiemIsar, VpntXetNghiemIsar, QWhere> {

20485 │ QueryBuilder<VpntXetNghiemIsar, VpntXetNghiemIsar, QAfterWhere> anyId() {

20486 │ return QueryBuilder.apply(this, (query) {

20487 │ return query.addWhereClause(const IdWhereClause.any());

20488 │ });

20489 │ }

20490 │ }

20491 │

20492 │ extension VpntXetNghiemIsarQueryWhere

20493 │ on QueryBuilder<VpntXetNghiemIsar, VpntXetNghiemIsar, QWhereClause> {

20494 │ QueryBuilder<VpntXetNghiemIsar, VpntXetNghiemIsar, QAfterWhereClause>

20495 │ idEqualTo(Id id) {

20496 │ return QueryBuilder.apply(this, (query) {

20497 │ return query.addWhereClause(IdWhereClause.between(

20498 │ lower: id,

20499 │ upper: id,

20500 │ ));

749 │ });

750 │ }

751 │ }

752 │

753 │ extension WardsIsarQueryObject

754 │ on QueryBuilder<WardsIsar, WardsIsar, QFilterCondition> {}

755 │

756 │ extension WardsIsarQueryLinks

757 │ on QueryBuilder<WardsIsar, WardsIsar, QFilterCondition> {}

758 │

759 │ extension WardsIsarQuerySortBy on QueryBuilder<WardsIsar, WardsIsar, QSortBy> {

760 │ QueryBuilder<WardsIsar, WardsIsar, QAfterSortBy> sortByMa() {

761 │ return QueryBuilder.apply(this, (query) {

762 │ return query.addSortBy(r'ma', Sort.asc);

763 │ });

764 │ }

765 │

766 │ QueryBuilder<WardsIsar, WardsIsar, QAfterSortBy> sortByMaDesc() {

767 │ return QueryBuilder.apply(this, (query) {

768 │ return query.addSortBy(r'ma', Sort.desc);

769 │ });

770 │ }

771 │

772 │ QueryBuilder<WardsIsar, WardsIsar, QAfterSortBy> sortByMahuyen() {

773 │ return QueryBuilder.apply(this, (query) {

774 │ return query.addSortBy(r'mahuyen', Sort.asc);

775 │ });

776 │ }

777 │

778 │ QueryBuilder<WardsIsar, WardsIsar, QAfterSortBy> sortByMahuyenDesc() {

779 │ return QueryBuilder.apply(this, (query) {

780 │ return query.addSortBy(r'mahuyen', Sort.desc);

781 │ });

782 │ }

783 │

784 │ QueryBuilder<WardsIsar, WardsIsar, QAfterSortBy> sortByTen() {

785 │ return QueryBuilder.apply(this, (query) {

786 │ return query.addSortBy(r'ten', Sort.asc);

787 │ });

788 │ }

789 │

790 │ QueryBuilder<WardsIsar, WardsIsar, QAfterSortBy> sortByTenDesc() {

791 │ return QueryBuilder.apply(this, (query) {

792 │ return query.addSortBy(r'ten', Sort.desc);

793 │ });

794 │ }

795 │ }

796 │

797 │ extension WardsIsarQuerySortThenBy

798 │ on QueryBuilder<WardsIsar, WardsIsar, QSortThenBy> {

799 │ QueryBuilder<WardsIsar, WardsIsar, QAfterSortBy> thenById() {

800 │ return QueryBuilder.apply(this, (query) {

801 │ return query.addSortBy(r'id', Sort.asc);

802 │ });

803 │ }

804 │

805 │ QueryBuilder<WardsIsar, WardsIsar, QAfterSortBy> thenByIdDesc() {

806 │ return QueryBuilder.apply(this, (query) {

807 │ return query.addSortBy(r'id', Sort.desc);

808 │ });

809 │ }

810 │

811 │ QueryBuilder<WardsIsar, WardsIsar, QAfterSortBy> thenByMa() {

812 │ return QueryBuilder.apply(this, (query) {

813 │ return query.addSortBy(r'ma', Sort.asc);

814 │ });

815 │ }

816 │

817 │ QueryBuilder<WardsIsar, WardsIsar, QAfterSortBy> thenByMaDesc() {

818 │ return QueryBuilder.apply(this, (query) {

819 │ return query.addSortBy(r'ma', Sort.desc);

820 │ });

821 │ }

822 │

823 │ QueryBuilder<WardsIsar, WardsIsar, QAfterSortBy> thenByMahuyen() {

824 │ return QueryBuilder.apply(this, (query) {

825 │ return query.addSortBy(r'mahuyen', Sort.asc);

826 │ });

827 │ }

828 │

829 │ QueryBuilder<WardsIsar, WardsIsar, QAfterSortBy> thenByMahuyenDesc() {

830 │ return QueryBuilder.apply(this, (query) {

831 │ return query.addSortBy(r'mahuyen', Sort.desc);

832 │ });

833 │ }

834 │

835 │ QueryBuilder<WardsIsar, WardsIsar, QAfterSortBy> thenByTen() {

836 │ return QueryBuilder.apply(this, (query) {

837 │ return query.addSortBy(r'ten', Sort.asc);

838 │ });

839 │ }

840 │

841 │ QueryBuilder<WardsIsar, WardsIsar, QAfterSortBy> thenByTenDesc() {

842 │ return QueryBuilder.apply(this, (query) {

843 │ return query.addSortBy(r'ten', Sort.desc);

844 │ });

845 │ }

846 │ }

847 │

848 │ extension WardsIsarQueryWhereDistinct

849 │ on QueryBuilder<WardsIsar, WardsIsar, QDistinct> {

850 │ QueryBuilder<WardsIsar, WardsIsar, QDistinct> distinctByMa(

851 │ {bool caseSensitive = true}) {

852 │ return QueryBuilder.apply(this, (query) {

853 │ return query.addDistinctBy(r'ma', caseSensitive: caseSensitive);

854 │ });

855 │ }

856 │

857 │ QueryBuilder<WardsIsar, WardsIsar, QDistinct> distinctByMahuyen(

858 │ {bool caseSensitive = true}) {

859 │ return QueryBuilder.apply(this, (query) {

860 │ return query.addDistinctBy(r'mahuyen', caseSensitive: caseSensitive);

861 │ });

862 │ }

863 │

864 │ QueryBuilder<WardsIsar, WardsIsar, QDistinct> distinctByTen(

865 │ {bool caseSensitive = true}) {

866 │ return QueryBuilder.apply(this, (query) {

867 │ return query.addDistinctBy(r'ten', caseSensitive: caseSensitive);

868 │ });

869 │ }

870 │ }

871 │

872 │ extension WardsIsarQueryProperty

873 │ on QueryBuilder<WardsIsar, WardsIsar, QQueryProperty> {

874 │ QueryBuilder<WardsIsar, int, QQueryOperations> idProperty() {

875 │ return QueryBuilder.apply(this, (query) {

876 │ return query.addPropertyName(r'id');

877 │ });

878 │ }

879 │

880 │ QueryBuilder<WardsIsar, String, QQueryOperations> maProperty() {

881 │ return QueryBuilder.apply(this, (query) {

882 │ return query.addPropertyName(r'ma');

883 │ });

884 │ }

885 │

886 │ QueryBuilder<WardsIsar, String, QQueryOperations> mahuyenProperty() {

887 │ return QueryBuilder.apply(this, (query) {

888 │ return query.addPropertyName(r'mahuyen');

889 │ });

890 │ }

891 │

892 │ QueryBuilder<WardsIsar, String, QQueryOperations> tenProperty() {

893 │ return QueryBuilder.apply(this, (query) {

894 │ return query.addPropertyName(r'ten');

895 │ });

896 │ }

897 │ }

898 │

899 │ // \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

900 │ // JsonSerializableGenerator

901 │ // \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

902 │

903 │ \_$WardsModelImpl \_$$WardsModelImplFromJson(Map<String, dynamic> json) =>

904 │ \_$WardsModelImpl(

905 │ ma: json['ma'] as String,

906 │ ten: json['ten'] as String,

907 │ mahuyen: json['mahuyen'] as String,

908 │ );

909 │

910 │ Map<String, dynamic> \_$$WardsModelImplToJson(\_$WardsModelImpl instance) =>

911 │ <String, dynamic>{

912 │ 'ma': instance.ma,

913 │ 'ten': instance.ten,

914 │ 'mahuyen': instance.mahuyen,

915 │ };

────────────────────────────────────────────────────────────

**📄 web\_view\_bloc.dart (DART, 19 lines)**

1 │ // import 'package:bloc/bloc.dart';

2 │ // import 'package:flutter\_inappwebview/flutter\_inappwebview.dart';

3 │ // import 'package:freezed\_annotation/freezed\_annotation.dart';

4 │

5 │ // part 'web\_view\_event.dart';

6 │ // part 'web\_view\_state.dart';

7 │ // part 'web\_view\_bloc.freezed.dart';

8 │

9 │ // class WebViewBloc extends Bloc<WebViewEvent, WebViewState> {

10 │ // //final WebViewRepository \_repository;

11 │ // // ignore: unused\_field

12 │ // InAppWebViewController? \_webViewController;

13 │ // WebViewBloc() : super(const WebViewState()) {

14 │ // on<WebViewEvent>((event, emit) {

15 │ // // implement event handler

16 │ // });

17 │ // }

18 │ // }

19 │ // //

────────────────────────────────────────────────────────────

**📄 web\_view\_event.dart (DART, 33 lines)**

1 │ // part of 'web\_view\_bloc.dart';

2 │

3 │ // @freezed

4 │ // class WebViewEvent with \_$WebViewEvent {

5 │ // const factory WebViewEvent.initialized({

6 │ // required String url,

7 │ // @Default(false) bool useSystemBrowser,

8 │ // }) = \_WebViewInitialized;

9 │

10 │ // const factory WebViewEvent.urlChanged(String url) = \_WebViewUrlChanged;

11 │

12 │ // const factory WebViewEvent.loadStarted() = \_WebViewLoadStarted;

13 │

14 │ // const factory WebViewEvent.loadCompleted() = \_WebViewLoadCompleted;

15 │

16 │ // const factory WebViewEvent.progressUpdated(double progress) =

17 │ // \_WebViewProgressUpdated;

18 │

19 │ // const factory WebViewEvent.navigationStateChanged({

20 │ // required bool canGoBack,

21 │ // required bool canGoForward,

22 │ // }) = \_WebViewNavigationStateChanged;

23 │

24 │ // const factory WebViewEvent.goBack() = \_WebViewGoBack;

25 │

26 │ // const factory WebViewEvent.goForward() = \_WebViewGoForward;

27 │

28 │ // const factory WebViewEvent.reload() = \_WebViewReload;

29 │

30 │ // const factory WebViewEvent.stopLoading() = \_WebViewStopLoading;

31 │

32 │ // const factory WebViewEvent.openInBrowser() = \_WebViewOpenInBrowser;

33 │ // }

────────────────────────────────────────────────────────────

**📄 web\_view\_repository.dart (DART, 8 lines)**

1 │ // // features/web\_view/domain/repositories/web\_view\_repository.dart

2 │ // import 'package:flutter\_inappwebview/flutter\_inappwebview.dart';

3 │

4 │ // abstract class WebViewRepository {

5 │ // Future<void> openWebContent(String url, {bool useSystemBrowser = false});

6 │ // Future<void> setController(InAppWebViewController controller);

7 │ // Future<void> dispose();

8 │ // }

────────────────────────────────────────────────────────────

**📄 web\_view\_repository\_impl.dart (DART, 25 lines)**

1 │ // // features/web\_view/data/repositories/web\_view\_repository\_impl.dart

2 │ // import 'package:dkkb\_app/4\_app/service/in\_app\_webview/web\_view\_repository.dart';

3 │ // import 'package:flutter\_inappwebview/flutter\_inappwebview.dart';

4 │

5 │ // class WebViewRepositoryImpl implements WebViewRepository {

6 │ // final WebViewService \_webViewService;

7 │

8 │ // WebViewRepositoryImpl(this.\_webViewService);

9 │

10 │ // @override

11 │ // Future<void> openWebContent(String url, {bool useSystemBrowser = false}) {

12 │ // return \_webViewService.openWebContent(url,

13 │ // useSystemBrowser: useSystemBrowser);

14 │ // }

15 │

16 │ // @override

17 │ // Future<void> setController(InAppWebViewController controller) {

18 │ // return \_webViewService.setController(controller);

19 │ // }

20 │

21 │ // @override

22 │ // Future<void> dispose() {

23 │ // return \_webViewService.dispose();

24 │ // }

25 │ // }

────────────────────────────────────────────────────────────

**📄 web\_view\_service\_impl.dart (DART, 159 lines)**

1 │ // import 'dart:io';

2 │ // import 'package:dkkb\_app/5\_app/color/app\_colors.dart';

3 │ // import 'package:dkkb\_app/5\_app/constant/constant.dart';

4 │ // import 'package:dkkb\_app/8\_service/in\_app\_webview/chrome\_safari\_service.dart';

5 │ // import 'package:dkkb\_app/8\_service/in\_app\_webview/i\_web\_view\_service.dart';

6 │ // import 'package:flutter/material.dart';

7 │ // import 'package:flutter\_inappwebview/flutter\_inappwebview.dart';

8 │ // import 'package:url\_launcher/url\_launcher.dart';

9 │

10 │ // class WebViewServiceImpl implements IWebViewService {

11 │ // final ChromeSafariBrowser \_systemBrowser = MyChromeSafariBrowser();

12 │ // InAppWebViewController? \_controller;

13 │

14 │ // void setController(InAppWebViewController controller) {

15 │ // \_controller = controller;

16 │ // }

17 │

18 │ // @override

19 │ // Future<void> openWebContent(String url,

20 │ // {bool useSystemBrowser = false}) async {

21 │ // if (useSystemBrowser) {

22 │ // if (Platform.isAndroid) {

23 │ // await \_openWithCustomTabs(url);

24 │ // } else if (Platform.isIOS) {

25 │ // await \_openWithSafariViewController(url);

26 │ // }

27 │ // } else {

28 │ // await \_controller?.loadUrl(

29 │ // urlRequest: URLRequest(

30 │ // url: WebUri(url),

31 │ // headers: {

32 │ // 'Accept': 'text/html,application/xhtml+xml,application/xml',

33 │ // 'User-Agent': 'Mozilla/5.0',

34 │ // },

35 │ // ),

36 │ // );

37 │ // }

38 │ // }

39 │

40 │ // Future<void> \_openWithCustomTabs(String url) async {

41 │ // // Tạo animation resources

42 │ // final List<AndroidResource> startAnimations = [

43 │ // AndroidResource(

44 │ // name: AppConstant.animSlideUp,

45 │ // defType: 'anim',

46 │ // defPackage: AppConstant.packageName,

47 │ // ),

48 │ // AndroidResource(

49 │ // name: AppConstant.animSlideDown,

50 │ // defType: 'anim',

51 │ // defPackage: AppConstant.packageName,

52 │ // ),

53 │ // ];

54 │

55 │ // final settings = ChromeSafariBrowserSettings(

56 │ // // Cấu hình cơ bản

57 │ // presentationStyle: ModalPresentationStyle.FULL\_SCREEN,

58 │ // enableUrlBarHiding: true,

59 │ // barCollapsingEnabled: true,

60 │ // showTitle: true,

61 │

62 │ // // Màu sắc

63 │ // toolbarBackgroundColor: AppColors.textColorBlue,

64 │ // navigationBarColor: AppColors.textColorBlue,

65 │ // navigationBarDividerColor: Colors.white,

66 │ // secondaryToolbarColor: Colors.white,

67 │

68 │ // // Animations

69 │ // startAnimations: startAnimations,

70 │

71 │ // // Browser behavior

72 │ // keepAliveEnabled: true,

73 │ // isSingleInstance: false,

74 │ // alwaysUseBrowserUI: false);

75 │

76 │ // // Thêm custom menu items

77 │ // \_systemBrowser.addMenuItems([

78 │ // ChromeSafariBrowserMenuItem(

79 │ // id: 1,

80 │ // label: 'Làm mới',

81 │ // onClick: (url, title) async {

82 │ // await \_controller?.reload();

83 │ // }),

84 │ // ChromeSafariBrowserMenuItem(

85 │ // id: 2,

86 │ // label: 'Mở trong trình duyệt',

87 │ // onClick: (url, title) async {

88 │ // if (url != null) {

89 │ // final Uri uri = Uri.parse(url.toString());

90 │ // await ChromeSafariBrowser.isAvailable().then((available) async {

91 │ // if (available) {

92 │ // await ChromeSafariBrowser.clearWebsiteData();

93 │ // await launchUrl(uri, mode: LaunchMode.externalApplication);

94 │ // }

95 │ // });

96 │ // }

97 │ // }),

98 │ // ]);

99 │

100 │ // await \_systemBrowser.open(

101 │ // settings: settings,

102 │ // url: WebUri(url),

103 │ // headers: {

104 │ // 'User-Agent': 'Mozilla/5.0',

105 │ // 'X-Title': 'Bệnh viện Hồng Hưng', // Thử thêm custom header

106 │ // },

107 │ // );

108 │ // }

109 │

110 │ // Future<void> \_openWithSafariViewController(String url) async {

111 │ // await \_systemBrowser.open(

112 │ // settings: ChromeSafariBrowserSettings(

113 │ // presentationStyle: ModalPresentationStyle.FULL\_SCREEN,

114 │ // enableUrlBarHiding: true,

115 │ // barCollapsingEnabled: true,

116 │ // showTitle: true,

117 │ // dismissButtonStyle: DismissButtonStyle.CLOSE,

118 │ // preferredBarTintColor: AppColors.textColorBlue,

119 │ // preferredControlTintColor: Colors.white,

120 │ // entersReaderIfAvailable: false,

121 │ // alwaysUseBrowserUI: false,

122 │ // ),

123 │ // url: WebUri(url),

124 │ // );

125 │ // }

126 │

127 │ // Future<bool> canGoBack() async {

128 │ // return await \_controller?.canGoBack() ?? false;

129 │ // }

130 │

131 │ // Future<bool> canGoForward() async {

132 │ // return await \_controller?.canGoForward() ?? false;

133 │ // }

134 │

135 │ // Future<void> goBack() async {

136 │ // if (await canGoBack()) {

137 │ // await \_controller?.goBack();

138 │ // }

139 │ // }

140 │

141 │ // Future<void> goForward() async {

142 │ // if (await canGoForward()) {

143 │ // await \_controller?.goForward();

144 │ // }

145 │ // }

146 │

147 │ // Future<void> reload() async {

148 │ // await \_controller?.reload();

149 │ // }

150 │

151 │ // Future<void> stopLoading() async {

152 │ // await \_controller?.stopLoading();

153 │ // }

154 │

155 │ // @override

156 │ // void dispose() {

157 │ // \_controller = null;

158 │ // }

159 │ // }

────────────────────────────────────────────────────────────

**📄 web\_view\_state.dart (DART, 21 lines)**

1 │ // part of 'web\_view\_bloc.dart';

2 │

3 │ // @freezed

4 │ // class WebViewState with \_$WebViewState {

5 │ // const factory WebViewState({

6 │ // @Default(0.0) double progress,

7 │ // @Default(true) bool isLoading,

8 │ // @Default('') String currentUrl,

9 │ // @Default(false) bool canGoBack,

10 │ // @Default(false) bool canGoForward,

11 │ // @Default(WebViewStatus.initial) WebViewStatus status,

12 │ // String? error,

13 │ // }) = \_WebViewState;

14 │ // }

15 │

16 │ // enum WebViewStatus {

17 │ // initial,

18 │ // loading,

19 │ // loaded,

20 │ // error,

21 │ // }

────────────────────────────────────────────────────────────

**📄 week\_selector.dart (DART, 54 lines)**

1 │ import 'package:flutter/material.dart';

2 │

3 │ class WeekSelector extends StatelessWidget {

4 │ final String selectedWeek;

5 │ final Function(String) onWeekChanged;

6 │

7 │ const WeekSelector({

8 │ super.key,

9 │ required this.selectedWeek,

10 │ required this.onWeekChanged,

11 │ });

12 │

13 │ @override

14 │ Widget build(BuildContext context) {

15 │ return Row(

16 │ children: [

17 │ Expanded(

18 │ child: \_buildWeekButton('1', 'Tuần này'),

19 │ ),

20 │ const SizedBox(width: 12),

21 │ Expanded(

22 │ child: \_buildWeekButton('0', 'Tuần sau'),

23 │ ),

24 │ ],

25 │ );

26 │ }

27 │

28 │ Widget \_buildWeekButton(String value, String text) {

29 │ bool isSelected = selectedWeek == value;

30 │ return InkWell(

31 │ onTap: () => onWeekChanged(value),

32 │ child: Container(

33 │ padding: const EdgeInsets.symmetric(vertical: 12),

34 │ decoration: BoxDecoration(

35 │ color: isSelected ? Colors.teal : Colors.white,

36 │ borderRadius: BorderRadius.circular(30),

37 │ border: Border.all(

38 │ color: Colors.teal,

39 │ width: isSelected ? 0 : 1,

40 │ ),

41 │ ),

42 │ child: Center(

43 │ child: Text(

44 │ text,

45 │ style: TextStyle(

46 │ color: isSelected ? Colors.white : Colors.teal,

47 │ fontWeight: FontWeight.w600,

48 │ ),

49 │ ),

50 │ ),

51 │ ),

52 │ );

53 │ }

54 │ }

────────────────────────────────────────────────────────────

**📄 widget.dart (DART, 10 lines)**

1 │ export 'infor\_user\_address\_form.dart';

2 │

3 │ export 'infor\_user\_cccd\_input.dart';

4 │ export 'infor\_user\_gender\_radio.dart';

5 │ export 'infor\_user\_ho\_ten\_input.dart';

6 │ export 'infor\_user\_birthday\_input.dart';

7 │ export 'infor\_user\_phone\_input.dart';

8 │ export 'infor\_user\_submit\_button.dart';

9 │ export 'infor\_user\_address\_selection.dart';

10 │ export 'infor\_user\_bhyt\_input.dart';

────────────────────────────────────────────────────────────

**📄 widget\_test.dart (DART, 29 lines)**

1 │ // This is a basic Flutter widget test.

2 │ //

3 │ // To perform an interaction with a widget in your test, use the WidgetTester

4 │ // utility in the flutter\_test package. For example, you can send tap and scroll

5 │ // gestures. You can also use WidgetTester to find child widgets in the widget

6 │ // tree, read text, and verify that the values of widget properties are correct.

7 │

8 │ import 'package:flutter/material.dart';

9 │ import 'package:flutter\_test/flutter\_test.dart';

10 │ import 'package:dkkb\_app/my\_app.dart';

11 │

12 │ void main() {

13 │ testWidgets('Counter increments smoke test', (WidgetTester tester) async {

14 │ // Build our app and trigger a frame.

15 │ await tester.pumpWidget(const MyApp());

16 │

17 │ // Verify that our counter starts at 0.

18 │ expect(find.text('0'), findsOneWidget);

19 │ expect(find.text('1'), findsNothing);

20 │

21 │ // Tap the '+' icon and trigger a frame.

22 │ await tester.tap(find.byIcon(Icons.add));

23 │ await tester.pump();

24 │

25 │ // Verify that our counter has incremented.

26 │ expect(find.text('0'), findsNothing);

27 │ expect(find.text('1'), findsOneWidget);

28 │ });

29 │ }

────────────────────────────────────────────────────────────

**📄 widgets.dart (DART, 17 lines)**

1 │ export 'appbar\_custom.dart';

2 │ export 'background\_custom.dart';

3 │ export 'biometric\_switch.dart';

4 │ export 'SosicalIconButton.dart';

5 │ export 'copyright\_row.dart';

6 │ export 'custom\_adress\_elevated\_button.dart';

7 │ export 'custom\_button.dart';

8 │ export 'custom\_input\_field.dart';

9 │ export 'custom\_input\_field\_pass.dart';

10 │ export 'language\_switch.dart';

11 │ export 'language\_switch2.dart';

12 │ export 'login\_appbar.dart';

13 │ export 'password\_input\_field.dart';

14 │ export 'res\_password\_input\_field.dart';

15 │ export 'ten\_bac\_si\_container.dart';

16 │ export 'ten\_chuyen\_khoa\_container.dart';

17 │ export 'ten\_loai\_dich\_vu\_container.dart';

────────────────────────────────────────────────────────────

**📄 widgets.dart (DART, 10 lines)**

1 │ export 'cardThamMy/card\_chua\_thuc\_hien.dart';

2 │ export 'cardThamMy/card\_da\_thuc\_hien.dart';

3 │ export 'cardThamMy/card\_goi\_dich\_vu.dart';

4 │ export 'cardThamMy/custom\_card\_shape\_painter.dart';

5 │ export 'cardThamMy/place\_info.dart';

6 │ export 'card\_infor\_benh\_nhan\_tham\_my.dart';

7 │ export 'tab\_chua\_thuc\_hien.dart';

8 │ export 'tab\_da\_thuc\_hien.dart';

9 │ export 'tab\_goi\_dich\_vu.dart';

10 │ export 'tab\_view\_thammy.dart';

────────────────────────────────────────────────────────────

**📄 widgets.dart (DART, 8 lines)**

1 │ export 'card\_infor\_benh\_nhan.dart';

2 │ export 'error\_widget.dart';

3 │ export '../screens/list\_benh\_nhan\_tiem\_chung.dart';

4 │ export 'loading\_widget.dart';

5 │ export 'tab\_chua\_tiem.dart';

6 │ export 'tab\_da\_tiem.dart';

7 │ export 'tabview\_mui\_tiem.dart';

8 │ export 'tiem\_chung\_list.dart';

────────────────────────────────────────────────────────────

**📄 x\_quang\_repository.dart (DART, 68 lines)**

1 │ import 'package:dkkb\_app/1\_models/api\_response/api\_response.dart';

2 │ import 'package:dkkb\_app/1\_models/imaging\_result/app\_model/imaging\_result\_model.dart';

3 │ import 'package:dkkb\_app/1\_models/imaging\_result/isar\_model/imaging\_result\_isar.dart';

4 │ import 'package:dkkb\_app/3\_repositories/cdha/base\_imaging\_repository.dart';

5 │ import 'package:dkkb\_app/8\_service/extension/api\_response\_extension.dart';

6 │

7 │ // Abstract interface for X-Quang repository

8 │ abstract class IXQuangRepository {

9 │ Future<ApiResponse<ImagingResponseModel>> danhSachKetQuaXQuang({

10 │ required String patientId,

11 │ bool forceRefresh = false,

12 │ });

13 │ Future<ApiResponse<ImagingResponseModel>> refreshKetQuaXQuang(

14 │ String patientId);

15 │ Future<ApiResponse<ImagingResponseModel>> syncKetQuaXQuang(String patientId);

16 │ }

17 │

18 │ // Concrete implementation of X-Quang repository

19 │ class XQuangRepository extends BaseImagingRepository

20 │ implements IXQuangRepository {

21 │ XQuangRepository({

22 │ required super.apiService,

23 │ required super.imagingDao,

24 │ required super.syncTimeManager,

25 │ required super.connectivity,

26 │ });

27 │

28 │ @override

29 │ Future<ApiResponse<ImagingResponseModel>> fetchFromApi(

30 │ String patientId) async {

31 │ final response = await apiService.fetchdanhSachKetQuaXQuang(patientId);

32 │ return response.mapToObject(

33 │ ImagingResponseModel.fromJson,

34 │ defaultValue: ImagingResponseModel(),

35 │ );

36 │ }

37 │

38 │ @override

39 │ ImagingType get imagingType => ImagingType.xQuang;

40 │

41 │ @override

42 │ String get imagingName => 'X-Quang';

43 │

44 │ @override

45 │ Future<ApiResponse<ImagingResponseModel>> danhSachKetQuaXQuang({

46 │ required String patientId,

47 │ bool forceRefresh = false,

48 │ }) {

49 │ return getImagingResults(

50 │ patientId: patientId,

51 │ forceRefresh: forceRefresh,

52 │ );

53 │ }

54 │

55 │ @override

56 │ Future<ApiResponse<ImagingResponseModel>> refreshKetQuaXQuang(

57 │ String patientId,

58 │ ) {

59 │ return refresh(patientId);

60 │ }

61 │

62 │ @override

63 │ Future<ApiResponse<ImagingResponseModel>> syncKetQuaXQuang(

64 │ String patientId,

65 │ ) {

66 │ return sync(patientId);

67 │ }

68 │ }

────────────────────────────────────────────────────────────

**📄 x\_ray\_content.dart (DART, 112 lines)**

1 │ import 'package:dkkb\_app/1\_models/imaging\_result/app\_model/imaging\_result\_model.dart';

2 │ import 'package:dkkb\_app/4\_presention/Pages/PDF/CDHA/x\_ray/x\_ray\_html\_cover.dart';

3 │ import 'package:dkkb\_app/4\_presention/Pages/PDF/base/pdf\_page\_dimensions.dart';

4 │ import 'package:dkkb\_app/5\_app/utils/PDF\_Utils/html\_cover.dart';

5 │ import 'package:dkkb\_app/5\_app/utils/PDF\_Utils/text\_custom.dart';

6 │ import 'package:dkkb\_app/commons.dart';

7 │

8 │ import 'package:pdf/widgets.dart' as pw;

9 │

10 │ pw.Widget buildXRayContent({

11 │ required List<Uint8List> optimizedImages,

12 │ required ImagingResultModel result,

13 │ required String resultTitle,

14 │ required pw.Font ttf,

15 │ required pw.Font ttfBold,

16 │ required pw.Font ttfItalic,

17 │ }) {

18 │ return pw.Expanded(

19 │ child: pw.Column(

20 │ crossAxisAlignment: pw.CrossAxisAlignment.start,

21 │ children: [

22 │ // Phần tiêu đề

23 │ pw.Container(

24 │ padding: const pw.EdgeInsets.symmetric(vertical: 8),

25 │ child: pw.Center(

26 │ child: TextCustom(

27 │ title: resultTitle,

28 │ ttf: ttfBold,

29 │ fontSize: PDFPageDimensions.fontSizes['title']!.sp,

30 │ ),

31 │ ),

32 │ ),

33 │

34 │ // \* Phần nội dung chính - thay đổi từ height cố định sang Expanded

35 │ // ! Bỏ Container height cố định 350.h, thay bằng Expanded

36 │ pw.Expanded(

37 │ child: pw.Column(

38 │ crossAxisAlignment: pw.CrossAxisAlignment.start,

39 │ children: [

40 │ // \* Phần mô tả và kết luận

41 │ pw.Expanded(

42 │ child: pw.Row(

43 │ mainAxisAlignment: pw.MainAxisAlignment.start,

44 │ crossAxisAlignment: pw.CrossAxisAlignment.start,

45 │ children: [

46 │ // \* Cột nội dung văn bản

47 │ pw.Expanded(

48 │ child: pw.Padding(

49 │ padding: pw.EdgeInsets.only(left: 5.w),

50 │ child: pw.Column(

51 │ crossAxisAlignment: pw.CrossAxisAlignment.start,

52 │ children: [

53 │ // Phần mô tả

54 │ TextCustom(

55 │ title: '\* Mô tả (Description):',

56 │ ttf: ttfBold,

57 │ fontSize: PDFPageDimensions

58 │ .fontSizes['sectionHeader']!.sp,

59 │ ),

60 │ pw.SizedBox(height: 8),

61 │ pw.Expanded(

62 │ flex: 3, // ? Cho phần mô tả chiếm nhiều space hơn

63 │ child: pw.Padding(

64 │ padding: pw.EdgeInsets.only(left: 8.w),

65 │ child: XRayHtmlCover(

66 │ ttfBold: ttf,

67 │ fontSize: PDFPageDimensions

68 │ .fontSizes['content']!.sp,

69 │ htmlString: result.result ?? '',

70 │ ttf: ttf,

71 │ ),

72 │ ),

73 │ ),

74 │

75 │ pw.SizedBox(height: 16),

76 │

77 │ // Phần kết luận

78 │ TextCustom(

79 │ title: '\* Kết luận (Conclusion):',

80 │ ttf: ttfBold,

81 │ fontSize: PDFPageDimensions

82 │ .fontSizes['sectionHeader']!.sp,

83 │ ),

84 │ pw.SizedBox(height: 8),

85 │ pw.Expanded(

86 │ flex: 2,

87 │ child: pw.Padding(

88 │ padding: pw.EdgeInsets.only(left: 8.w),

89 │ child: HTMLCover(

90 │ // Thay đổi từ TextCustom sang HTMLCover

91 │ ttfBold: ttf,

92 │ fontSize: PDFPageDimensions

93 │ .fontSizes['content']!.sp,

94 │ htmlString: result.conclusion ?? '',

95 │ ttf: ttf,

96 │ ),

97 │ ),

98 │ ),

99 │ ],

100 │ ),

101 │ ),

102 │ ),

103 │ ],

104 │ ),

105 │ ),

106 │ ],

107 │ ),

108 │ ),

109 │ ],

110 │ ),

111 │ );

112 │ }

────────────────────────────────────────────────────────────

**📄 x\_ray\_html\_cover.dart (DART, 110 lines)**

1 │ import 'package:pdf/widgets.dart' as pw;

2 │ // ignore: library\_prefixes

3 │ import 'package:html/parser.dart' as htmlParser;

4 │

5 │ class XRayHtmlCover extends pw.StatelessWidget {

6 │ final String htmlString;

7 │ final pw.Font ttf;

8 │ final pw.Font ttfBold;

9 │ final double fontSize;

10 │

11 │ XRayHtmlCover({

12 │ required this.htmlString,

13 │ required this.ttf,

14 │ required this.ttfBold,

15 │ required this.fontSize,

16 │ });

17 │

18 │ @override

19 │ pw.Widget build(pw.Context context) {

20 │ final document = htmlParser.parse(htmlString);

21 │

22 │ // Tìm vùng chụp

23 │ String? imagingArea;

24 │ final pTags = document.getElementsByTagName('p');

25 │ if (pTags.isNotEmpty) {

26 │ imagingArea = pTags.first.text.trim();

27 │ }

28 │

29 │ // Lấy dữ liệu từ table

30 │ final List<Map<String, String>> tableData = [];

31 │ final rows = document.getElementsByTagName('tr');

32 │

33 │ for (var row in rows) {

34 │ final cells = row.getElementsByTagName('td');

35 │ if (cells.length >= 2) {

36 │ String key = cells[0].text.trim();

37 │ String value = cells[1].text.trim();

38 │

39 │ // Loại bỏ dấu - ở đầu key nếu có

40 │ if (key.startsWith('-')) {

41 │ key = key.substring(1).trim();

42 │ }

43 │

44 │ tableData.add({

45 │ 'key': key,

46 │ 'value': value,

47 │ });

48 │ }

49 │ }

50 │

51 │ return pw.Column(

52 │ crossAxisAlignment: pw.CrossAxisAlignment.start,

53 │ children: [

54 │ // Tiêu đề vùng chụp

55 │ if (imagingArea != null)

56 │ pw.Container(

57 │ margin: const pw.EdgeInsets.only(bottom: 8),

58 │ child: pw.Text(

59 │ imagingArea,

60 │ style: pw.TextStyle(

61 │ font: ttfBold,

62 │ fontSize: fontSize + 1, // Slightly larger than content

63 │ ),

64 │ ),

65 │ ),

66 │

67 │ // Nội dung khám

68 │ ...tableData.map((data) => pw.Container(

69 │ margin: const pw.EdgeInsets.only(bottom: 4),

70 │ child: pw.Row(

71 │ crossAxisAlignment: pw.CrossAxisAlignment.start,

72 │ children: [

73 │ // Key (label)

74 │ pw.Container(

75 │ width: 100, // Fixed width for labels

76 │ margin: const pw.EdgeInsets.only(right: 8),

77 │ child: pw.Text(

78 │ data['key']!,

79 │ style: pw.TextStyle(

80 │ font: ttfBold,

81 │ fontSize: fontSize,

82 │ ),

83 │ ),

84 │ ),

85 │ // Value

86 │ pw.Expanded(

87 │ child: pw.Text(

88 │ data['value']!,

89 │ style: pw.TextStyle(

90 │ font: ttf,

91 │ fontSize: fontSize,

92 │ ),

93 │ textAlign: pw.TextAlign.justify,

94 │ ),

95 │ ),

96 │ ],

97 │ ),

98 │ )),

99 │ ],

100 │ );

101 │ }

102 │

103 │ // Helper method để clean text

104 │ String \_cleanText(String text) {

105 │ return text

106 │ .replaceAll(RegExp(r'<[^>]\*>'), '')

107 │ .replaceAll('&nbsp;', ' ')

108 │ .trim();

109 │ }

110 │ }

────────────────────────────────────────────────────────────

**📄 x\_ray\_pdf\_generator.dart (DART, 144 lines)**

1 │ // pdf\_generators/ultrasound\_pdf\_generator.dart

2 │ import 'package:dkkb\_app/1\_models/imaging\_result/extention.dart';

3 │ import 'package:dkkb\_app/1\_models/imaging\_result/app\_model/imaging\_result\_model.dart';

4 │ import 'package:dkkb\_app/4\_presention/Pages/PDF/CDHA/x\_ray/x\_ray\_content.dart';

5 │ import 'package:dkkb\_app/4\_presention/Pages/PDF/base/base\_pdf\_generator.dart';

6 │ import 'package:dkkb\_app/4\_presention/Pages/PDF/base/pdf\_page\_dimensions.dart';

7 │ import 'package:dkkb\_app/4\_presention/Pages/PDF/base/pdf\_utils.dart';

8 │

9 │ import 'dart:developer' as developer;

10 │ import 'package:dkkb\_app/commons.dart';

11 │ import 'package:pdf/pdf.dart';

12 │ import 'package:pdf/widgets.dart' as pw;

13 │

14 │ // \* Class UltrasoundPDFGenerator dùng để tạo file PDF cho kết quả siêu âm

15 │ // ? Kế thừa từ BasePDFGenerator để tận dụng các phương thức cơ bản của việc tạo PDF

16 │ class XRayPdfGenerator extends BasePDFGenerator<ImagingResponseModel> {

17 │ // \* Phương thức chính để tạo file PDF từ dữ liệu siêu âm

18 │ // @param format: Định dạng trang PDF

19 │ // @param data: Dữ liệu kết quả siêu âm

20 │ // @param ttf: Font chữ thường

21 │ // @param ttfBold: Font chữ đậm

22 │ // @param ttfItalic: Font chữ nghiêng

23 │ // @param signature: Chữ ký bác sĩ (tuỳ chọn)

24 │ // @return Future<Uint8List>: Dữ liệu PDF dạng binary

25 │ @override

26 │ Future<Uint8List> generatePDF({

27 │ required PdfPageFormat format,

28 │ required ImagingResponseModel data,

29 │ required pw.Font ttf,

30 │ required pw.Font ttfBold,

31 │ required pw.Font ttfItalic,

32 │ pw.ImageProvider? signature,

33 │ }) async {

34 │ // Log số lượng kết quả để debug

35 │ developer.log('Generating PDF with ${data.imagingResults.length} results');

36 │

37 │ // Khởi tạo document và theme

38 │ // ? Đặt tên document là 'Siêu âm' để dễ nhận biết khi mở file

39 │ final doc = pw.Document(title: 'Siêu âm');

40 │ final pageTheme = await myPageTheme(format);

41 │

42 │ // ! Cần đảm bảo load tất cả hình ảnh trước khi tạo PDF

43 │ await initializeImages();

44 │

45 │ // Duyệt qua từng kết quả siêu âm để tạo các trang PDF

46 │ // ? Sử dụng asMap() để lấy được cả index và giá trị của mỗi kết quả

47 │ for (var entry in data.imagingResults.asMap().entries) {

48 │ final index = entry.key;

49 │ final result = entry.value;

50 │

51 │ try {

52 │ // \* Load trước các hình ảnh cần thiết cho mỗi trang

53 │ final images = await preloadImages(result.resultImageUrls);

54 │

55 │ // \* Thêm một trang mới vào document

56 │ doc.addPage(

57 │ pw.Page(

58 │ pageTheme: pageTheme,

59 │ build: (context) {

60 │ // ? Sử dụng Container bao ngoài để kiểm soát chiều cao tổng thể

61 │ return pw.Container(

62 │ height: PDFPageDimensions.contentHeight,

63 │ child: pw.Column(

64 │ children: [

65 │ // \* Phần nội dung chính của trang

66 │ pw.Expanded(

67 │ child: pw.Column(

68 │ crossAxisAlignment: pw.CrossAxisAlignment.start,

69 │ children: [

70 │ // 1. Phần header

71 │ buildHeader(ttf: ttf),

72 │ pw.SizedBox(height: 5),

73 │

74 │ // 2. Thông tin bệnh nhân

75 │ // ! Chiều cao cố định để đảm bảo layout thống nhất

76 │ pw.Container(

77 │ height: PDFPageDimensions.getPatientInfoHeight(),

78 │ child: buildPersonalInfo(

79 │ hoTen: data.patientInfo?.fullName ?? '',

80 │ ngaySinh: data.patientInfo?.dateOfBirth ?? '',

81 │ gioiTinh: data.patientInfo?.gender ?? '',

82 │ diaChi: data.patientInfo?.address ?? '',

83 │ code: data.patientInfo?.patientCode ?? '',

84 │ vpntInfo: data.prescriptionInfo,

85 │ ngayChiDinh: result.orderDate ?? '',

86 │ ttf: ttf,

87 │ ttfBold: ttfBold,

88 │ ttfItalic: ttfItalic,

89 │ ),

90 │ ),

91 │

92 │ // 3. Nội dung kết quả siêu âm

93 │ pw.Container(

94 │ height: PDFPageDimensions.getMainContentHeight(),

95 │ child: buildXRayContent(

96 │ // ? Kiểm tra index để tránh lỗi khi truy cập orderData

97 │ resultTitle: index < data.orderData.length

98 │ ? data.orderData[index].resultTitle ?? ''

99 │ : '',

100 │ optimizedImages: images,

101 │ result: result,

102 │ ttf: ttf,

103 │ ttfBold: ttfBold,

104 │ ttfItalic: ttfItalic,

105 │ ),

106 │ ),

107 │

108 │ // 4. Thông tin bác sĩ và chữ ký

109 │ pw.Container(

110 │ height: PDFPageDimensions.getBottomHeight() \* 0.7,

111 │ child: buildBottomInfo(

112 │ ngaygiotraketqua: result.resultDateTime ?? '',

113 │ code: result.imageUrl ?? '',

114 │ signatureImage: signature,

115 │ bacSiThucHien: result.performingDoctor ?? '',

116 │ ttf: ttf,

117 │ ttfBold: ttfBold,

118 │ ttfItalic: ttfItalic,

119 │ ),

120 │ ),

121 │ ],

122 │ ),

123 │ ),

124 │ // Footer của trang

125 │ buildFooter(ttf: ttf),

126 │ ],

127 │ ),

128 │ );

129 │ },

130 │ ),

131 │ );

132 │ } catch (e) {

133 │ // ! Xử lý lỗi cho từng trang riêng biệt để không ảnh hưởng đến các trang khác

134 │ developer.log('Error generating page for result: $e');

135 │ continue;

136 │ }

137 │ }

138 │

139 │ // Thêm tính năng nén PDF để giảm dung lượng file

140 │ // Thêm watermark cho PDF nếu cần

141 │

142 │ return doc.save();

143 │ }

144 │ }

────────────────────────────────────────────────────────────

**📄 x\_ray\_pdf\_preview\_page.dart (DART, 38 lines)**

1 │ // \* Import các package cần thiết cho việc tạo và xem trước PDF kết quả siêu âm

2 │ import 'package:dkkb\_app/1\_models/imaging\_result/app\_model/imaging\_result\_model.dart';

3 │ import 'package:dkkb\_app/4\_presention/Pages/PDF/CDHA/x\_ray/x\_ray\_pdf\_generator.dart';

4 │

5 │ import 'package:dkkb\_app/4\_presention/Pages/PDF/base/base\_pdf\_generator.dart';

6 │ import 'package:dkkb\_app/4\_presention/Pages/PDF/base/base\_pdf\_preview\_page.dart';

7 │

8 │ // \* Widget hiển thị trang xem trước PDF kết quả siêu âm

9 │ // ? Kế thừa từ BasePDFPreviewPage để tận dụng các chức năng cơ bản của trang xem trước PDF

10 │ // @param data: Dữ liệu kết quả siêu âm cần hiển thị

11 │ class XRayPdfPreviewPage extends BasePDFPreviewPage<ImagingResponseModel> {

12 │ // \* Constructor của class

13 │ // ! Phải truyền data khi khởi tạo widget này

14 │ // ? Title được set cứng là "Kết quả siêu âm" vì đây là trang preview kết quả siêu âm

15 │ const XRayPdfPreviewPage({

16 │ super.key,

17 │ required super.data,

18 │ }) : super(

19 │ title: "Kết quả X-quang",

20 │ );

21 │

22 │ // \* Override method tạo PDF generator

23 │ // ? Mỗi loại PDF cần có generator riêng, ở đây dùng UltrasoundPDFGenerator

24 │ // @return Trả về instance của UltrasoundPDFGenerator để tạo file PDF

25 │ @override

26 │ BasePDFGenerator<ImagingResponseModel> createPDFGenerator() =>

27 │ XRayPdfGenerator();

28 │

29 │ // \* Override method lấy tên file PDF

30 │ // ? Đặt tên file mặc định là 'sieu\_am.pdf'

31 │ // ! Tên file nên để tiếng Việt không dấu để tránh lỗi encoding

32 │ // @return Tên file PDF khi người dùng tải về

33 │ @override

34 │ String getFileName() => 'x\_quang.pdf';

35 │ }

36 │

37 │ // TODO: Có thể thêm các tùy chọn cho phép người dùng đặt tên file tùy ý

38 │ // TODO: Thêm các tùy chọn xem trước như zoom, rotate, etc.

────────────────────────────────────────────────────────────

**📄 xet\_nghiem\_content.dart (DART, 305 lines)**

1 │ // ignore\_for\_file: constant\_identifier\_names

2 │

3 │ import 'dart:math';

4 │

5 │ import 'package:dkkb\_app/1\_models/laboratory\_result/laboratory\_result\_model.dart';

6 │ import 'package:dkkb\_app/4\_presention/Pages/PDF/base/pdf\_page\_dimensions.dart';

7 │ import 'package:pdf/pdf.dart';

8 │ import 'package:pdf/widgets.dart' as pw;

9 │ import 'package:dkkb\_app/5\_app/utils/PDF\_Utils/text\_custom.dart';

10 │ import 'package:flutter\_screenutil/flutter\_screenutil.dart';

11 │

12 │ pw.Widget buildXetNghiemContent({

13 │ required String resultTitle,

14 │ required String tenLoaiDV,

15 │ required LaboratoryResultModel data,

16 │ required pw.Font ttf,

17 │ required pw.Font ttfBold,

18 │ required pw.Font ttfItalic,

19 │ required Map<String, List<dynamic>> currentResults,

20 │ }) {

21 │ const double MAX\_ROW\_HEIGHT = 15.0;

22 │ const double HEADER\_SPACE = 60.0;

23 │ // Tính toán kích thước dựa trên số lượng items

24 │ final totalItems =

25 │ currentResults.values.fold(0, (sum, list) => sum + list.length);

26 │

27 │ // Tính toán fontSize và spacing dựa trên số lượng items

28 │ double calculateFontSize() {

29 │ if (totalItems <= 10) {

30 │ return PDFPageDimensions.fontSizes['content']! \* 1.1; // Giảm xuống 90%

31 │ }

32 │ if (totalItems <= 20) {

33 │ return PDFPageDimensions.fontSizes['content']! \* 1.1; // Giảm xuống 80%

34 │ }

35 │ if (totalItems <= 30) {

36 │ return PDFPageDimensions.fontSizes['content']! \* 1.1; // Giảm xuống 70%

37 │ }

38 │ if (totalItems <= 40) {

39 │ return PDFPageDimensions.fontSizes['content']! \* 0.6; // Thêm mức 60%

40 │ }

41 │ return PDFPageDimensions.fontSizes['content']! \* 0.5; // Giảm xuống 50%

42 │ }

43 │

44 │ double calculateSpacing() {

45 │ if (totalItems <= 10) return 0.5.h; // Giảm từ 5.h

46 │ if (totalItems <= 20) return 0.5.h; // Giảm từ 3.h

47 │ if (totalItems <= 30) return 0.5.h; // Giảm từ 2.h

48 │ if (totalItems <= 40) return 0.8.h; // Thêm mức mới

49 │ return 0.5.h; // Giảm minimum spacing

50 │ }

51 │

52 │ final adjustedFontSize = calculateFontSize();

53 │ final adjustedSpacing = calculateSpacing();

54 │

55 │ // Tính chiều cao khả dụng

56 │ final availableHeight = PDFPageDimensions.getMainContentHeightXN();

57 │ final itemHeight =

58 │ min((availableHeight - HEADER\_SPACE) / totalItems, MAX\_ROW\_HEIGHT);

59 │

60 │ return pw.Column(

61 │ crossAxisAlignment: pw.CrossAxisAlignment.start,

62 │ children: [

63 │ // Phần tiêu đề

64 │ pw.Container(

65 │ padding: pw.EdgeInsets.symmetric(vertical: adjustedSpacing),

66 │ child: pw.Center(

67 │ child: TextCustomCenter(

68 │ title: resultTitle,

69 │ ttf: ttfBold,

70 │ fontSize: PDFPageDimensions.fontSizes['title']!.sp,

71 │ ),

72 │ ),

73 │ ),

74 │

75 │ // Header của bảng kết quả

76 │ pw.Container(

77 │ height: 40, // Fixed height for header

78 │ child: pw.Row(

79 │ mainAxisAlignment: pw.MainAxisAlignment.spaceBetween,

80 │ children: [

81 │ pw.Expanded(

82 │ flex: 12,

83 │ child: pw.Column(

84 │ mainAxisSize: pw.MainAxisSize.min,

85 │ crossAxisAlignment: pw.CrossAxisAlignment.start,

86 │ children: [

87 │ TextCustom(

88 │ title: 'Tên xét nghiệm',

89 │ ttf: ttfBold,

90 │ fontSize: adjustedFontSize),

91 │ pw.SizedBox(height: adjustedSpacing),

92 │ TextCustom(

93 │ title: 'Parameters',

94 │ ttf: ttfItalic,

95 │ fontSize: adjustedFontSize),

96 │ ],

97 │ ),

98 │ ),

99 │ pw.Expanded(

100 │ flex: 3,

101 │ child: pw.Column(

102 │ mainAxisSize: pw.MainAxisSize.min,

103 │ crossAxisAlignment: pw.CrossAxisAlignment.center,

104 │ children: [

105 │ TextCustomCenter(

106 │ title: 'Kết quả',

107 │ ttf: ttfBold,

108 │ fontSize: adjustedFontSize),

109 │ pw.SizedBox(height: adjustedSpacing),

110 │ TextCustomCenter(

111 │ title: 'Result',

112 │ ttf: ttfItalic,

113 │ fontSize: adjustedFontSize),

114 │ ],

115 │ ),

116 │ ),

117 │ // Thêm cột trống cho H/L (không có header)

118 │ pw.Expanded(

119 │ flex: 1,

120 │ child: pw.Column(

121 │ mainAxisSize: pw.MainAxisSize.min,

122 │ crossAxisAlignment: pw.CrossAxisAlignment.center,

123 │ children: [

124 │ pw.SizedBox(

125 │ height: adjustedFontSize \* 2), // Để trống phần header

126 │ ],

127 │ ),

128 │ ),

129 │ pw.Expanded(

130 │ flex: 3,

131 │ child: pw.Column(

132 │ mainAxisSize: pw.MainAxisSize.min,

133 │ crossAxisAlignment: pw.CrossAxisAlignment.center,

134 │ children: [

135 │ TextCustomCenter(

136 │ title: 'Đơn vị',

137 │ ttf: ttfBold,

138 │ fontSize: adjustedFontSize),

139 │ pw.SizedBox(height: adjustedSpacing),

140 │ TextCustomCenter(

141 │ title: 'Unit',

142 │ ttf: ttfItalic,

143 │ fontSize: adjustedFontSize),

144 │ ],

145 │ ),

146 │ ),

147 │ pw.Expanded(

148 │ flex: 4,

149 │ child: pw.Column(

150 │ mainAxisSize: pw.MainAxisSize.min,

151 │ crossAxisAlignment: pw.CrossAxisAlignment.center,

152 │ children: [

153 │ TextCustomCenter(

154 │ title: 'Giá trị tham chiếu',

155 │ ttf: ttfBold,

156 │ fontSize: adjustedFontSize),

157 │ pw.SizedBox(height: adjustedSpacing),

158 │ TextCustomCenter(

159 │ title: 'Ref Range',

160 │ ttf: ttfItalic,

161 │ fontSize: adjustedFontSize),

162 │ ],

163 │ ),

164 │ ),

165 │ ],

166 │ ),

167 │ ),

168 │ // Nội dung từng nhóm xét nghiệm

169 │ pw.Expanded(

170 │ child: pw.Column(

171 │ crossAxisAlignment: pw.CrossAxisAlignment.start,

172 │ children: [

173 │ // Tên loại dịch vụ ở ngoài ListView

174 │ TextCustom(

175 │ title: tenLoaiDV,

176 │ ttf: ttfBold,

177 │ fontSize: adjustedFontSize,

178 │ ),

179 │ pw.SizedBox(height: adjustedSpacing),

180 │ pw.Expanded(

181 │ child: pw.ListView(

182 │ children: currentResults.entries.map(

183 │ (nhomKT) {

184 │ return pw.Container(

185 │ child: pw.Column(

186 │ crossAxisAlignment: pw.CrossAxisAlignment.start,

187 │ children: [

188 │ TextCustom(

189 │ title: nhomKT.key,

190 │ ttf: ttfBold,

191 │ fontSize: adjustedFontSize),

192 │ pw.SizedBox(

193 │ height: adjustedSpacing,

194 │ ),

195 │

196 │ // Các kết quả trong nhóm

197 │ ...nhomKT.value.map(

198 │ (ketQua) => pw.Container(

199 │ height: itemHeight,

200 │ child: pw.Column(

201 │ children: [

202 │ pw.Row(

203 │ children: [

204 │ pw.Expanded(

205 │ flex: 12,

206 │ child: TextCustom(

207 │ title: "${ketQua.tendv}",

208 │ ttf: ttf,

209 │ fontSize: adjustedFontSize,

210 │ ),

211 │ ),

212 │ // Sau đó sửa phần hiển thị giá trị

213 │ pw.Expanded(

214 │ flex: 3,

215 │ child: TextCustomCenter(

216 │ title: ketQua.giatri,

217 │ ttf: ttf,

218 │ fontSize: adjustedFontSize,

219 │ color: checkAbnormalValue(

220 │ ketQua.giatri,

221 │ ketQua.giatrithap,

222 │ ketQua.giatricao,

223 │ )

224 │ ? PdfColors.red

225 │ : null, // Set màu đỏ nếu giá trị bất thường

226 │ ),

227 │ ),

228 │ // Thêm cột mới cho H/L

229 │ pw.Expanded(

230 │ flex: 1,

231 │ child: TextCustomCenter(

232 │ title: ketQua.ketluan ??

233 │ '', // Giả sử ketluan là trường chứa H/L

234 │ ttf: ttfBold,

235 │ fontSize: adjustedFontSize,

236 │ ),

237 │ ),

238 │ pw.Expanded(

239 │ flex: 3,

240 │ child: TextCustomCenter(

241 │ title: ketQua.donvi,

242 │ ttf: ttf,

243 │ fontSize: adjustedFontSize,

244 │ ),

245 │ ),

246 │ pw.Expanded(

247 │ flex: 4,

248 │ child: TextCustomCenter(

249 │ title: ketQua.gioihan,

250 │ ttf: ttf,

251 │ fontSize: adjustedFontSize,

252 │ ),

253 │ ),

254 │ ],

255 │ ),

256 │ pw.Container(

257 │ height: adjustedSpacing,

258 │ child: pw.Divider(color: PdfColors.grey200),

259 │ ),

260 │ ],

261 │ ),

262 │ ),

263 │ ),

264 │ ],

265 │ ),

266 │ );

267 │ },

268 │ ).toList(),

269 │ ),

270 │ ),

271 │ // Thêm dòng chú thích ở đây, sau expanded content

272 │ pw.SizedBox(height: adjustedSpacing \* 2), // Thêm khoảng cách

273 │ pw.Container(

274 │ padding: pw.EdgeInsets.only(bottom: adjustedSpacing),

275 │ child: TextCustom(

276 │ title:

277 │ 'Xin đối chiếu kết quả xét nghiệm và tình trạng người bệnh, các kết quả c�...

278 │ ttf: ttfItalic, // Dùng font italic cho phần note

279 │ fontSize: 7.sp,

280 │ ),

281 │ ),

282 │ ],

283 │ ),

284 │ ),

285 │ ],

286 │ );

287 │ }

288 │

289 │ // Hàm kiểm tra giá trị bất thường

290 │ bool checkAbnormalValue(String value, String min, String max) {

291 │ try {

292 │ double? val = double.tryParse(value);

293 │ double? minVal = double.tryParse(min);

294 │ double? maxVal = double.tryParse(max);

295 │

296 │ if (val == null) return false;

297 │

298 │ if (minVal != null && val < minVal) return true;

299 │ if (maxVal != null && val > maxVal) return true;

300 │

301 │ return false;

302 │ } catch (e) {

303 │ return false;

304 │ }

305 │ }

────────────────────────────────────────────────────────────

**📄 xet\_nghiem\_pdf\_generator.dart (DART, 315 lines)**

1 │ // pdf\_generators/ultrasound\_pdf\_generator.dart

2 │ // ignore\_for\_file: prefer\_interpolation\_to\_compose\_strings

3 │

4 │ import 'package:dkkb\_app/1\_models/laboratory\_result/laboratory\_result\_model.dart';

5 │ import 'package:dkkb\_app/4\_presention/Pages/PDF/base/base\_pdf\_generator.dart';

6 │ import 'package:dkkb\_app/4\_presention/Pages/PDF/base/pdf\_page\_dimensions.dart';

7 │ import 'package:dkkb\_app/4\_presention/Pages/PDF/base/pdf\_utils.dart';

8 │ import 'package:dkkb\_app/4\_presention/Pages/PDF/xet\_nghiem/xet\_nghiem\_content.dart';

9 │ import 'package:http/http.dart' as http;

10 │ import 'dart:developer' as developer;

11 │ import 'package:dkkb\_app/commons.dart';

12 │ import 'package:pdf/pdf.dart';

13 │ import 'package:pdf/widgets.dart' as pw;

14 │

15 │ // \* Class UltrasoundPDFGenerator dùng để tạo file PDF cho kết quả siêu âm

16 │ // ? Kế thừa từ BasePDFGenerator để tận dụng các phương thức cơ bản của việc tạo PDF

17 │ class XetNghiemPdfGenerator extends BasePDFGenerator<LaboratoryResultModel> {

18 │ // Hàm mới để tải hình ảnh chữ ký từ URL

19 │ Future<Uint8List?> \_loadSignatureImage(String signatureJson) async {

20 │ try {

21 │ // Parse JSON string

22 │ String processedJson = signatureJson.replaceAll('\\\\', '\\');

23 │ Map<String, dynamic> signatureInfo = jsonDecode(processedJson);

24 │

25 │ String signatureLink = signatureInfo['link'] ?? '';

26 │

27 │ // Kiểm tra link trống

28 │ if (signatureLink.isEmpty) {

29 │ return null;

30 │ }

31 │

32 │ // Thêm tham số view=1 để hiển thị trực tiếp

33 │ signatureLink = signatureLink + '?view=1';

34 │

35 │ // Tải hình ảnh

36 │ final response = await http.get(Uri.parse(signatureLink));

37 │

38 │ if (response.statusCode == 200) {

39 │ return response.bodyBytes;

40 │ } else {

41 │ developer.log(

42 │ 'Không thể tải hình ảnh chữ ký, mã lỗi: ${response.statusCode}');

43 │ return null;

44 │ }

45 │ } catch (e) {

46 │ developer.log('Lỗi khi tải hình ảnh chữ ký: $e');

47 │ return null;

48 │ }

49 │ }

50 │

51 │ // Trong class XetNghiemPdfGenerator, thêm hàm này để trích xuất ngayky

52 │ String \_extractNgayKy(String signatureJson) {

53 │ try {

54 │ // Parse JSON string

55 │ String processedJson = signatureJson.replaceAll('\\\\', '\\');

56 │ Map<String, dynamic> signatureInfo = jsonDecode(processedJson);

57 │

58 │ // Lấy giá trị ngayky từ JSON

59 │ String ngayKy = signatureInfo['ngayky'] ?? '';

60 │ return ngayKy;

61 │ } catch (e) {

62 │ developer.log('Lỗi khi trích xuất ngayky: $e');

63 │ return '';

64 │ }

65 │ }

66 │

67 │ @override

68 │ Future<Uint8List> generatePDF({

69 │ required PdfPageFormat format,

70 │ required LaboratoryResultModel data,

71 │ required pw.Font ttf,

72 │ required pw.Font ttfBold,

73 │ required pw.Font ttfItalic,

74 │ pw.ImageProvider? signature,

75 │ }) async {

76 │ final doc = pw.Document(title: 'Xét Nghiệm');

77 │ final pageTheme = await myPageTheme(format);

78 │

79 │ // ! Cần đảm bảo load tất cả hình ảnh trước khi tạo PDF

80 │ await initializeImages();

81 │

82 │ // Nhóm kết quả theo tenloaidv

83 │ final Map<String, Map<String, List<dynamic>>> groupedResults = {};

84 │

85 │ // Phân loại dữ liệu

86 │ for (var item in data.chitiet) {

87 │ if (!groupedResults.containsKey(item.tenloaidv)) {

88 │ groupedResults[item.tenloaidv] = {};

89 │ }

90 │

91 │ if (!groupedResults[item.tenloaidv]!.containsKey(item.tennhomktxn)) {

92 │ groupedResults[item.tenloaidv]![item.tennhomktxn] = [];

93 │ }

94 │

95 │ groupedResults[item.tenloaidv]![item.tennhomktxn]!.add(item);

96 │ }

97 │

98 │ // Tạo một Map để lưu trữ thông tin khoa và các thông tin khác cho mỗi loại dịch vụ

99 │ final Map<String, Map<String, dynamic>> serviceInfo = {};

100 │

101 │ // Map để lưu thông tin orderID, ngayChiDinh và vpntInfo cho mỗi loại dịch vụ

102 │ final Map<String, Map<String, dynamic>> personalInfo = {};

103 │

104 │ // Map để lưu trữ hình ảnh chữ ký cho mỗi loại dịch vụ

105 │ final Map<String, pw.ImageProvider?> signatureImages = {};

106 │

107 │ // Lặp qua tất cả kết quả để lấy thông tin và tải chữ ký

108 │ for (var ketqua in data.ketqua) {

109 │ if (!serviceInfo.containsKey(ketqua.tenloaidv)) {

110 │ // Lưu thông tin dịch vụ

111 │ serviceInfo[ketqua.tenloaidv] = {

112 │ 'tenkhoa': ketqua.tenkhoa,

113 │ 'ngaynhanmau': ketqua.ngaynhanmau,

114 │ 'nguoinhanmau': ketqua.nguoinhanmau,

115 │ 'nguoiduyet': ketqua.nguoiduyet,

116 │ 'ngayduyet': ketqua.ngayduyet,

117 │ 'chukynguoiduyet': ketqua.chukynguoiduyet,

118 │ 'ngaytruongkhoaky': \_extractNgayKy(ketqua.chukytruongkhoa),

119 │ };

120 │

121 │ // Lưu thông tin cá nhân

122 │ personalInfo[ketqua.tenloaidv] = {

123 │ 'orderid': ketqua.orderid,

124 │ 'ngaychidinh': ketqua.ngaychidinh,

125 │ 'vpntInfo': data.vpnt, // Assuming vpnt is same for all services

126 │ };

127 │

128 │ // Tải và lưu hình ảnh chữ ký

129 │ if (ketqua.chukynguoiduyet.isNotEmpty) {

130 │ try {

131 │ final signatureBytes =

132 │ await \_loadSignatureImage(ketqua.chukytruongkhoa);

133 │ if (signatureBytes != null) {

134 │ signatureImages[ketqua.tenloaidv] =

135 │ pw.MemoryImage(signatureBytes);

136 │ }

137 │ } catch (e) {

138 │ developer

139 │ .log('Lỗi khi tải hình ảnh chữ ký cho ${ketqua.tenloaidv}: $e');

140 │ }

141 │ }

142 │ }

143 │ }

144 │

145 │ // Tạo trang PDF cho mỗi loại dịch vụ

146 │ for (var loaiDV in groupedResults.entries) {

147 │ try {

148 │ final info = serviceInfo[loaiDV.key] ?? serviceInfo.values.first;

149 │ final pInfo = personalInfo[loaiDV.key] ?? personalInfo.values.first;

150 │ final signatureImage = signatureImages[loaiDV.key];

151 │

152 │ doc.addPage(

153 │ pw.Page(

154 │ pageTheme: pageTheme,

155 │ build: (context) {

156 │ return pw.Container(

157 │ height: PDFPageDimensions.contentHeight,

158 │ child: pw.Column(

159 │ children: [

160 │ pw.Expanded(

161 │ child: pw.Column(

162 │ crossAxisAlignment: pw.CrossAxisAlignment.start,

163 │ children: [

164 │ // 1. Header

165 │ buildHeader(ttf: ttf),

166 │ pw.SizedBox(height: 5),

167 │

168 │ // 2. Thông tin bệnh nhân

169 │ pw.Container(

170 │ height: PDFPageDimensions.getPatientInfoHeightXN(),

171 │ child: buildPersonalInfoXN(

172 │ hoTen: data.benhnhan.tenbenhnhan,

173 │ ngaySinh: data.benhnhan.ngaysinh,

174 │ gioiTinh: data.benhnhan.gioitinh,

175 │ diaChi: data.benhnhan.diachi,

176 │ code: data.benhnhan.codebenhnhan,

177 │ orderID: pInfo['orderid'],

178 │ vpntInfo: pInfo['vpntInfo'],

179 │ ketqua: data.ketqua,

180 │ ngayChiDinh: pInfo['ngaychidinh'],

181 │ ttf: ttf,

182 │ ttfBold: ttfBold,

183 │ ttfItalic: ttfItalic,

184 │ ),

185 │ ),

186 │

187 │ // 3. Nội dung của loại dịch vụ cụ thể

188 │ pw.Container(

189 │ height: PDFPageDimensions.getMainContentHeightXN(),

190 │ child: buildXetNghiemContent(

191 │ resultTitle:

192 │ "KẾT QUẢ XÉT NGHIỆM - LABORATORY REPORT",

193 │ tenLoaiDV: loaiDV.key,

194 │ data: data,

195 │ ttf: ttf,

196 │ ttfBold: ttfBold,

197 │ ttfItalic: ttfItalic,

198 │ currentResults: loaiDV

199 │ .value, // Chỉ truyền kết quả của loại dịch vụ này

200 │ ),

201 │ ),

202 │

203 │ // 4. Thông tin bác sĩ và chữ ký

204 │ pw.Container(

205 │ height: PDFPageDimensions.getBottomHeightXN() \* 0.7,

206 │ child: buildBottomXetNghiemInfo(

207 │ khoa: info['tenkhoa'] ?? "Khoa Xét Nghiệm",

208 │ ngayNhanMau: info['ngaynhanmau'] ?? "",

209 │ nguoiNhanMau: info['nguoinhanmau'] ?? "",

210 │ kiemSoatBoi: info['nguoiduyet'] ?? "",

211 │ ngaytruongkhoaky: info['ngaytruongkhoaky'] ?? "",

212 │ chuKyJson:

213 │ info['chukynguoiduyet'], // Truyền JSON chữ ký

214 │ signatureImage:

215 │ signatureImage, // Truyền hình ảnh chữ ký đã tải

216 │ ttf: ttf,

217 │ ttfBold: ttfBold,

218 │ ttfItalic: ttfItalic,

219 │ fontSize: 8.sp,

220 │ ),

221 │ ),

222 │ ],

223 │ ),

224 │ ),

225 │ buildFooter(ttf: ttf),

226 │ ],

227 │ ),

228 │ );

229 │ },

230 │ ),

231 │ );

232 │ } catch (e) {

233 │ developer.log('Error generating page for ${loaiDV.key}: $e');

234 │ continue;

235 │ }

236 │ }

237 │

238 │ // Sau khi tạo xong các trang kết quả, kiểm tra và thêm trang hình ảnh từ lstcdha

239 │ for (var cdhaResult in data.lstcdha) {

240 │ try {

241 │ // Tải hình ảnh từ fileketqua của CĐHA

242 │ if (cdhaResult.fileketqua.isNotEmpty) {

243 │ final imageBytes = await preloadImages([cdhaResult.fileketqua]);

244 │

245 │ if (imageBytes.isNotEmpty) {

246 │ doc.addPage(

247 │ pw.Page(

248 │ pageTheme: pageTheme,

249 │ build: (context) {

250 │ return pw.Container(

251 │ height: PDFPageDimensions.contentHeight,

252 │ child: pw.Column(

253 │ children: [

254 │ buildHeader(ttf: ttf),

255 │

256 │ pw.Expanded(

257 │ child: pw.Center(

258 │ child: pw.Container(

259 │ constraints: pw.BoxConstraints(

260 │ maxWidth: PDFPageDimensions.contentWidth \* 0.95,

261 │ maxHeight:

262 │ PDFPageDimensions.contentHeight \* 0.7,

263 │ ),

264 │ child: pw.Image(

265 │ pw.MemoryImage(imageBytes.first),

266 │ fit: pw.BoxFit.contain,

267 │ ),

268 │ ),

269 │ ),

270 │ ),

271 │

272 │ // Thông tin CĐHA và bệnh nhân

273 │ pw.Container(

274 │ alignment: pw.Alignment.centerLeft,

275 │ padding: const pw.EdgeInsets.only(

276 │ left: 10,

277 │ bottom: 2,

278 │ ),

279 │ child: pw.Column(

280 │ crossAxisAlignment: pw.CrossAxisAlignment.start,

281 │ children: [

282 │ pw.Text(

283 │ '${data.benhnhan.tenbenhnhan} - ${data.benhnhan.codebenhnhan}',

284 │ style: pw.TextStyle(font: ttf, fontSize: 10),

285 │ ),

286 │ pw.Text(

287 │ 'Dịch vụ: ${cdhaResult.tendichvu}',

288 │ style: pw.TextStyle(font: ttf, fontSize: 10),

289 │ ),

290 │ pw.Text(

291 │ 'Ngày thực hiện: ${cdhaResult.ngaythuchien}',

292 │ style: pw.TextStyle(font: ttf, fontSize: 10),

293 │ ),

294 │ ],

295 │ ),

296 │ ),

297 │

298 │ buildFooter(ttf: ttf),

299 │ ],

300 │ ),

301 │ );

302 │ },

303 │ ),

304 │ );

305 │ }

306 │ }

307 │ } catch (e) {

308 │ developer.log('Error adding CĐHA image page: $e');

309 │ continue;

310 │ }

311 │ }

312 │

313 │ return doc.save();

314 │ }

315 │ }

────────────────────────────────────────────────────────────

**📄 xet\_nghiem\_pdf\_preview\_page.dart (DART, 38 lines)**

1 │ // \* Import các package cần thiết cho việc tạo và xem trước PDF kết quả siêu âm

2 │ import 'package:dkkb\_app/1\_models/laboratory\_result/laboratory\_result\_model.dart';

3 │ import 'package:dkkb\_app/4\_presention/Pages/PDF/base/base\_pdf\_generator.dart';

4 │ import 'package:dkkb\_app/4\_presention/Pages/PDF/base/base\_pdf\_preview\_page.dart';

5 │ import 'package:dkkb\_app/4\_presention/Pages/PDF/xet\_nghiem/xet\_nghiem\_pdf\_generator.dart';

6 │

7 │ // \* Widget hiển thị trang xem trước PDF kết quả siêu âm

8 │ // ? Kế thừa từ BasePDFPreviewPage để tận dụng các chức năng cơ bản của trang xem trước PDF

9 │ // @param data: Dữ liệu kết quả siêu âm cần hiển thị

10 │ class XetNghiemPdfPreviewPage

11 │ extends BasePDFPreviewPage<LaboratoryResultModel> {

12 │ // \* Constructor của class

13 │ // ! Phải truyền data khi khởi tạo widget này

14 │ // ? Title được set cứng là "Kết quả siêu âm" vì đây là trang preview kết quả siêu âm

15 │ const XetNghiemPdfPreviewPage({

16 │ super.key,

17 │ required super.data,

18 │ }) : super(

19 │ title: "Kết quả xét nghiệm",

20 │ );

21 │

22 │ // \* Override method tạo PDF generator

23 │ // ? Mỗi loại PDF cần có generator riêng, ở đây dùng UltrasoundPDFGenerator

24 │ // @return Trả về instance của UltrasoundPDFGenerator để tạo file PDF

25 │ @override

26 │ BasePDFGenerator<LaboratoryResultModel> createPDFGenerator() =>

27 │ XetNghiemPdfGenerator();

28 │

29 │ // \* Override method lấy tên file PDF

30 │ // ? Đặt tên file mặc định là 'sieu\_am.pdf'

31 │ // ! Tên file nên để tiếng Việt không dấu để tránh lỗi encoding

32 │ // @return Tên file PDF khi người dùng tải về

33 │ @override

34 │ String getFileName() => 'ket\_qua\_xet\_nghiem.pdf';

35 │ }

36 │

37 │ // TODO: Có thể thêm các tùy chọn cho phép người dùng đặt tên file tùy ý

38 │ // TODO: Thêm các tùy chọn xem trước như zoom, rotate, etc.