

Experiment No 3

Name : Kunal Pal

Div/Roll no : D15B/50

Lab : MAD & PWA Lab

Aim : To include icons, images, fonts in Flutter app

Icons: Utilize icons to enhance visual representation and navigation within your app. Consider using icon libraries like `flutter_icons` or `font_awesome_flutter` to access a wide range of icons related to fitness activities, such as running, cycling, weightlifting, and nutrition. You can use the `Icon` widget to display these icons directly in your app's UI, providing intuitive cues for users to understand actions or features.

Images: Integrate relevant images to complement your app's content and engage users visually. Include images of fitness equipment, healthy food options, workout poses, or progress charts to motivate and inspire users in their fitness journey. You can use the `Image` widget along with `AssetImage` or `NetworkImage` to display images stored locally or fetched from remote sources. Ensure that images are optimized for different screen sizes and resolutions to maintain performance and quality.

Fonts: Choose fonts that align with your app's branding and readability goals. Select fonts that are clear, legible, and appropriate for fitness-related content. Consider using Google Fonts or custom font files to access a variety of font styles suitable for headings, body text, and emphasis. You can specify fonts globally using the `Theme` widget or apply them selectively to individual text elements using the `TextStyle` property. Experiment with font weights, sizes, and styles to create a visually appealing and cohesive typographic hierarchy throughout your app.

By incorporating icons, images, and fonts effectively into your FitTrack Flutter app, you can enhance its visual appeal, usability, and overall user experience.

Code :

```
import 'package:flutter/material.dart';
import 'package:flutter/services.dart';
import
'package:flutter_localizations/flutter_localizations.dart';
import 'core/app_export.dart';

var globalMessengerKey = GlobalKey<ScaffoldMessengerState>();
void main() {
  WidgetsFlutterBinding.ensureInitialized();
  Future.wait([
    SystemChrome.setPreferredOrientations([
      DeviceOrientation.portraitUp,
    ]),
    PrefUtils().init()
  ]).then((value) {
    runApp(MyApp());
  });
}

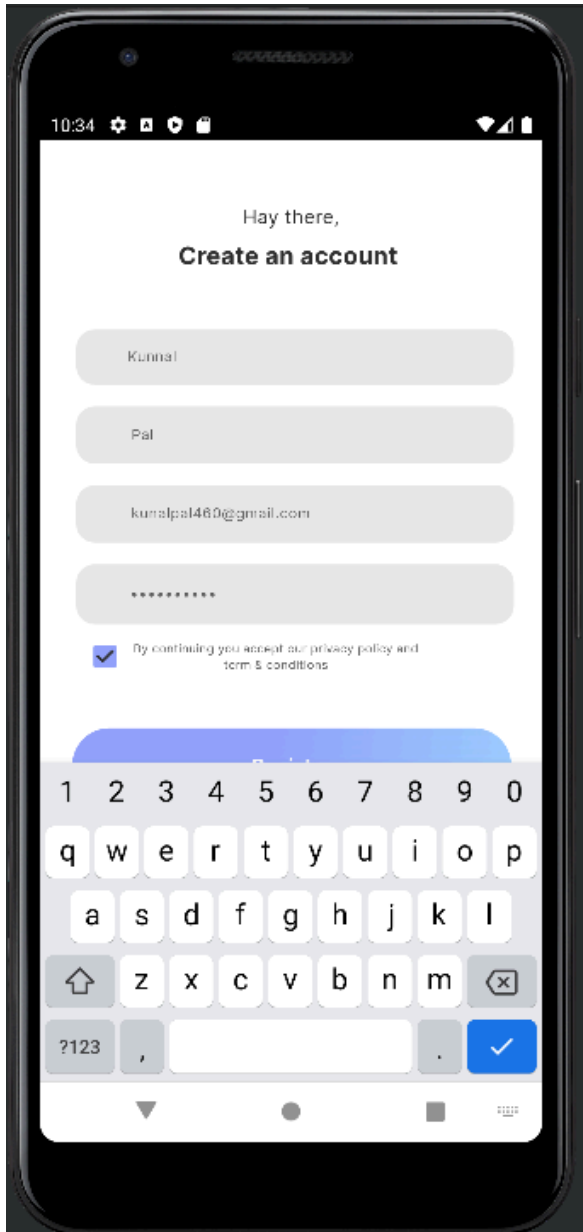
class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Sizer(
      builder: (context, orientation, deviceType) {
        return BlocProvider(
          create: (context) => ThemeBloc(
            ThemeState(
              themeType: PrefUtils().getThemeData(),
            ),
          ),
          child: BlocBuilder<ThemeBloc, ThemeState>(
            builder: (context, state) {
```

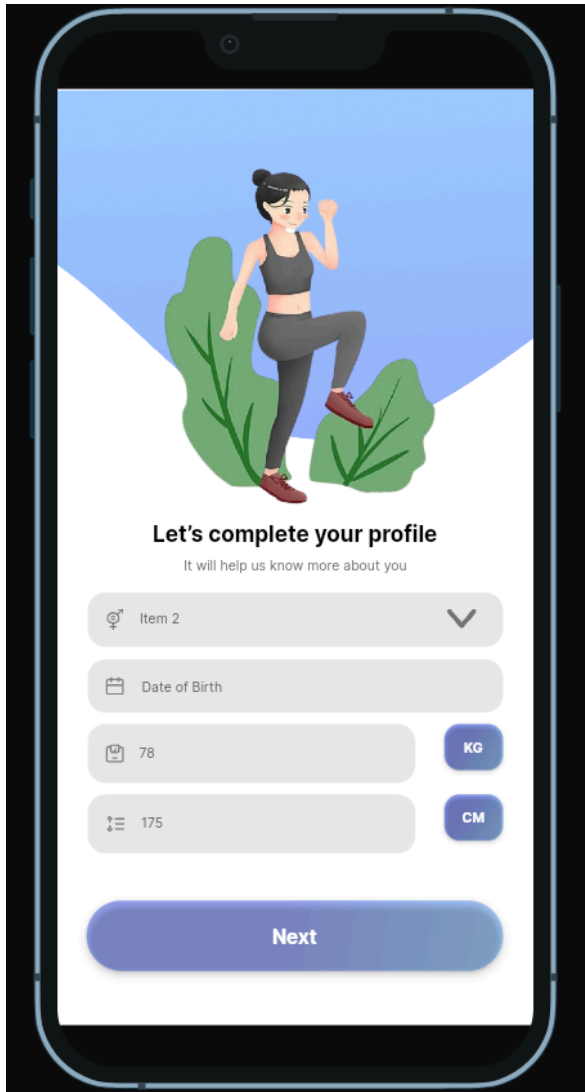
```

return MaterialApp(
  theme: theme,
  title: 'fitttrack',
  navigatorKey: NavigatorService.navigatorKey,
  debugShowCheckedModeBanner: false,
  localizationsDelegates: [
    AppLocalizationDelegate(),
    GlobalMaterialLocalizations.delegate,
    GlobalWidgetsLocalizations.delegate,
    GlobalCupertinoLocalizations.delegate,
  ],
  supportedLocales: [
    Locale(
      'en',
      '',
    ),
  ],
  initialRoute: AppRoutes.initialRoute,
  routes: AppRoutes.routes,
);
},
),
);
},
);
}
}

```

Output :





Conclusion : In Flutter, you've discovered how to jazz up your app with icons, images, and unique fonts. Icons are a breeze with the Icon widget, images shine using Image.asset, and custom fonts add flair via TextStyle. Together, they spruce up your app, making it visually appealing and user-friendly.