

Experiment 3

- **Aim:-** To include icons, images, fonts in Flutter app
- **Theory:-**

The main goal of this experiment is to successfully incorporate icons, images, and fonts into the Flutter clone app, ensuring proper visibility and seamless integration within the UI.

Implementation Steps

1. Adding Icons:
 - Used **Icons** class for default Material icons.
 - Added the **cupertino_icons** package for additional icon support.
 - Ensured icons were placed correctly within buttons, navigation bars, and other UI components.
2. Adding Images:
 - Initially faced issues where network images failed to load due to HTTP request errors.
 - Resolved the issue by using a different image source such as Pexels or Unsplash with proper URLs.
 - Ensured visibility by debugging asset paths and network permissions.
3. Adding Fonts:
 - Custom fonts were defined in the **pubspec.yaml** file.
 - Applied different fonts in the **TextStyle** property of text widgets.
 - Ensured proper rendering by verifying font files and YAML indentation.

Challenges and Solutions

- Issue: Images not loading (HTTP request failure).
Solution: Switched to alternative image sources like Unsplash and checked network permissions in **AndroidManifest.xml**.
- Issue: Icons not appearing.
Solution: Verified dependencies and used the correct package import for Cupertino icons.
- Issue: Fonts not rendering properly.
Solution: Checked YAML formatting, ensured font files were in the correct directory, and refreshed dependencies.

Code:-

```
import 'package:flutter/material.dart';

import '../widgets/turf_card.dart';

import 'booking_screen.dart';

class HomeScreen extends StatelessWidget {

  final List<Map<String, String>> turfs = [

    {

      'name': 'Green Field Arena',

      'location': 'Dadar West',

      'price': '1200',

      'imagePath': 'assets/img.png', // Added image path

    },

    {

      'name': 'Sports Paradise',

      'location': 'Thane West',

      'price': '800',

      'imagePath': 'assets/turf1.png', // Added image path

    },

    {

      'name': 'Football Ground',

      'location': 'Andheri',

      'price': '1000',

      'imagePath': 'assets/turf2.png', // Added image path
```

```
},  
];
```

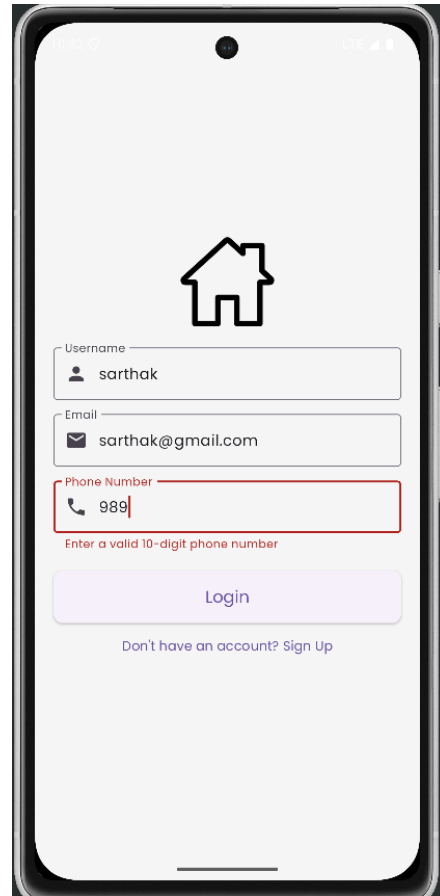
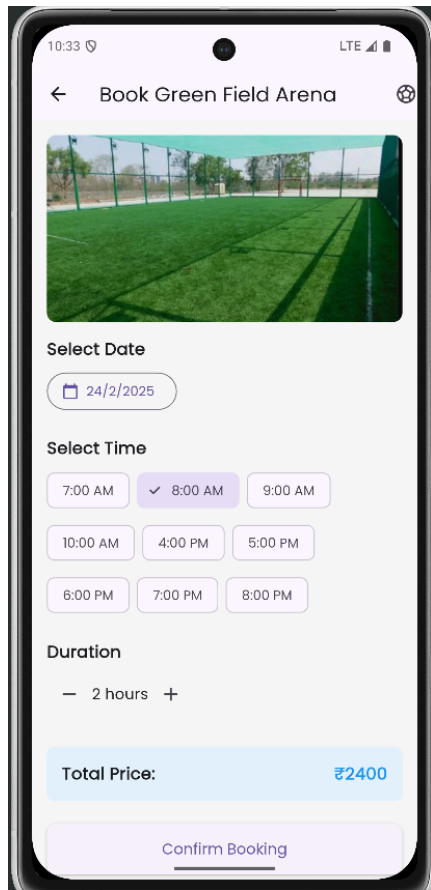
```
@override
```

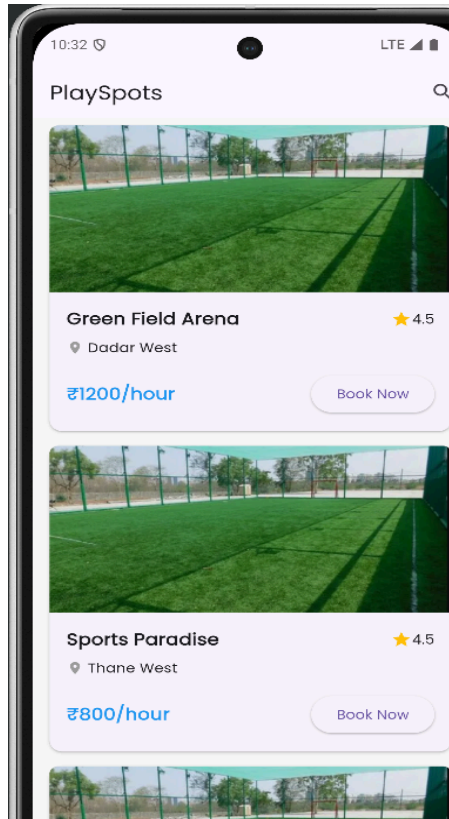
```
Widget build(BuildContext context) {
```

```
  return Scaffold(  
    appBar: AppBar(  
      title: Text('PlaySpots'),  
      actions: [  
        IconButton(  
          icon: Icon(Icons.search),  
          onPressed: () {},  
        ),  
      ],  
    ),  
    body: ListView.builder(  
      itemCount: turfs.length,  
      itemBuilder: (context, index) {  
        final turf = turfs[index];  
        return TurfCard(  
          name: turf['name']!,  
          location: turf['location']!,  
          price: turf['price']!,  
          imagePath: turf['imagePath']!, // Pass the image path here
```

```
onBook: () {  
    Navigator.push(  
        context,  
        MaterialPageRoute(  
            builder: (context) => BookingScreen(turfDetails: turf),  
        ),  
    );  
}  
);  
},  
),  
);  
}  
}
```

Output:-





- **Conclusion:-**

This experiment demonstrated how to effectively integrate icons, images, and fonts in a Flutter application. By handling network image loading errors, ensuring asset accessibility, and properly configuring fonts, we improved the UI/UX of the clone app. The enhancements made the application visually appealing and functional, aligning with modern mobile app design standards.