

Experiment 3

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

Theory:

Flutter provides various ways to include icons, images, and fonts in an application to improve visual appeal and branding. These assets can be loaded from different sources, such as:

- Flutter's built-in icons (Material Icons & Cupertino Icons).
- Local assets (images stored in the project directory).
- Network images (loaded from a URL).
- Custom fonts (TTF/OTF files added to the project).

Visual elements like icons, images, and fonts are essential for:

- Enhancing user interface aesthetic.
- Improving user experience through visual cues
- Maintaining brand consistency
- Enabling intuitive navigation

1. Icons in Flutter

Flutter provides two built-in icon sets:

- a. Material Icons (Android-style)
 - i. Extensive collection (~1,800 icons)
 - ii. Consistent with Google's design language
 - iii. Accessed via Icons class
- b. Cupertino Icons (iOS-style)
 - i. Matches Apple's design guidelines
 - ii. Accessed via CupertinoIcons class

Key Properties:

- a. Scalable vector graphics
- b. Customizable color and size
- c. Hardware-accelerated rendering

2. Image Handling

Flutter supports multiple image sources:

Types:

- a. Asset Images
 - i. Bundled with application
 - ii. Stored in project directory
 - iii. Requires declaration in pubspec.yaml
- b. Network Images

- i. Loaded from URLs
 - ii. Supports caching
 - iii. Requires internet permission
- c. Memory Images
 - i. Loaded from byte data
 - ii. Useful for dynamic content

Considerations:

- a. Memory management
 - b. Loading performance
 - c. Resolution adaptation
3. Font Management

Flutter allows custom font integration:

Implementation:

- a. Font files (TTF/OTF) added to project
- b. Configuration in pubspec.yaml
- c. Application through TextStyle

Features:

- a. Support for multiple font weights/styles
- b. Custom typography
- c. Platform-consistent rendering

Code:

1. Images:

Network image:

```
Image.network(
  awayLogoUrl,
  height: 40,
  width: 40,
  errorBuilder: (context, error, stackTrace) {
    return const Icon(
      Icons.error,
      size: 40,
      color: Colors.red,
    );
  },
  loadingBuilder: (context, child, loadingProgress) {
    if (loadingProgress == null) return child;
    return const SizedBox(
      height: 40,
      width: 40,
      child: Center(
        child: CircularProgressIndicator(
          strokeWidth: 2,
```

```

    ),
  ),
);
},
)

```



WAS

17-64

103

Image loaded from the network using the url provided for every team logo in the team repository.

Asset image:

Pubsec.yaml:

assets:

- assets/images/

ClipRect(

```

    borderRadius: const BorderRadius.only(
      topLeft: Radius.circular(16),
      topRight: Radius.circular(16),
    ),
    child: Stack(
      children: [
        // Match image
        Image.asset(
          'assets/images/news.jpg',
          width: double.infinity,
          height: 200, // Adjust image height
          fit: BoxFit.cover,
        ),
        // Overlaid headline at bottom-left
      ],
    ),
  ),
),

```

Top News



Lakers acquire Luka from Maverick for AD in blockbuster trade

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus et blandit velit. Aliquam convallis nisi et sapien lacinia, sit amet pellentesque leo malesuada.

2. Fonts and Text

```
Widget _buildDetailRow(String label, String value) {  
  return Padding(  
    padding: const EdgeInsets.symmetric(vertical: 2),  
    child: RichText(  
      text: TextSpan(  
        style: const TextStyle(color: Colors.black),  
        children: [  
          TextSpan(  
            text: '$label: ',  
            style: const TextStyle(fontWeight: FontWeight.bold),  
          ),  
          TextSpan(text: value),  
        ],  
      ),  
    ),  
  );  
}
```

Game ID: 20250411_MIL@DET

Season: 2025

Game Date: 20250411

Season Type: Regular Season

NBA.com: <https://www.nba.com/game/mil-vs-det-0022401171/box-score#box-score>

ESPN: https://www.espn.com/nba/boxscore/_/gameId/401705737

CBS: https://www.cbssports.com/nba/gametracker/boxscore/NBA_20250411_MIL@DET

3. Icons

```
class BlinkingDot extends StatefulWidget {
  const BlinkingDot({super.key});
```

```
  @override
  State<BlinkingDot> createState() => _BlinkingDotState();
}
```

```
class _BlinkingDotState extends State<BlinkingDot>
  with SingleTickerProviderStateMixin {
  late AnimationController _controller;
  late Animation<double> _animation;
```

```
  @override
  void initState() {
    super.initState();
    _controller = AnimationController(
      vsync: this,
      duration: const Duration(milliseconds: 800),
    )..repeat(reverse: true);
    _animation = Tween<double>(begin: 0.2, end: 1.0).animate(_controller);
  }
```

```
  @override
  Widget build(BuildContext context) {
    return FadeTransition(
      opacity: _animation,
      child: const Icon(
        Icons.circle,
        color: Colors.red,
        size: 12,
      ),
    );
  }
```

```
@override  
void dispose() {  
  _controller.dispose();  
  super.dispose();  
}  
}
```

Github link: <https://github.com/Rakshit5467/NBA-India>

Conclusion:

This experiment demonstrated how to effectively integrate icons, images, and fonts in a Flutter app to enhance UI/UX. Using network images (team logos), local assets (news images), custom text styling (RichText), and animated icons (blinking dot for live games), the app achieves visual consistency and responsiveness. Proper error handling (errorBuilder) and loading states (loadingBuilder) ensure a smooth user experience, while Material Icons and dynamic theming maintain platform-native aesthetics. The implementation highlights Flutter's flexibility in managing assets while optimizing performance and branding.