

Experiment No. 3

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

Theory:

To include icons, images, and fonts in a Flutter app, you need to understand the following core concepts related to asset management in Flutter. Here's the theory behind including these resources:

1. Assets in Flutter:

Assets are files or resources such as images, fonts, icons, or sounds that you include in your app and bundle within the app package. In Flutter, you can include these assets in your project and then use them in your app.

2. Adding Assets to pubspec.yaml:

In Flutter, you declare assets in the pubspec.yaml file. This is where you specify which assets should be bundled with your app during the build process.

Example for adding assets:

```
flutter:  
  assets:  
    - assets/images/  
    - assets/icons/
```

In this example, the images are stored in the assets/images directory, and icons in the assets/icons directory. You can also specify specific files instead of directories.

3. Including Images:

Flutter provides several ways to include images in your app, including network images, asset images, and file images. To use asset images, you reference them by their file path relative to the assets directory.

Example of including an asset image:

```
Image.asset('assets/images/my_image.png')
```

For this to work, the image (my_image.png) must be listed in the pubspec.yaml file under the flutter section, like this:

```
flutter:  
  assets:  
    - assets/images/my_image.png
```

4. Including Icons:

Flutter allows you to use custom icons in your app. You can add icon files (e.g., .png or .svg) to your assets folder and use them in the app. Alternatively, Flutter provides built-in icons via the Icons class.

Example of using an asset icon:
`Image.asset('assets/icons/my_icon.png')`

5. Including Fonts:

To include custom fonts, you place your font files (e.g., .ttf or .otf files) in a folder inside your assets directory. Then, you declare these fonts in the pubspec.yaml file and use them in your app.

Example of adding custom fonts in pubspec.yaml:

```
flutter:  
  fonts:  
    - family: CustomFont  
      fonts:  
        - asset: assets/fonts/CustomFont-Regular.ttf  
        - asset: assets/fonts/CustomFont-Bold.ttf
```

6. Font Weight and Style:

When specifying fonts, you can also define specific font weights and styles (like bold, italic) to support different text styles in your app.

7. Working with Icon Libraries:

While you can use custom icon files, Flutter also supports popular icon libraries like FontAwesome, MaterialIcons, etc. For example, Flutter's built-in Icons class provides access to the Material Design icons.

Example of using a Material icon:
`Icon(Icons.home)`

8. Caching and Optimization:

- **Images:** Flutter caches images, but you might want to use libraries like `cached_network_image` for better image loading and caching.

- **Fonts:** Custom fonts are loaded from assets when the app is first started, and they remain available for the lifecycle of the app.

9. SVG Images:

If you want to use vector-based images (like SVG), you can include them using packages like `flutter_svg`, which allows you to display scalable vector graphics (SVG files) in Flutter.

Example of including an SVG:

```
import 'package:flutter_svg/flutter_svg.dart';
SvgPicture.asset('assets/icons/my_icon.svg')
```

Code Snippets:

Moviedetailspage.dart:

```
import 'package:flutter/material.dart';
import 'package:cached_network_image/cached_network_image.dart';
import 'package:cloud_firestore/cloud_firestore.dart';
import '../utils/api_constants.dart';
import '../models/movie.dart';
import '../models/review.dart';
import '../services/movieDetails.dart';
import '../widgets/image.dart';

class MovieDetailsPage extends StatefulWidget {
  final String movieId;
  const MovieDetailsPage({Key? key, required this.movieId}) : super(key: key);
  @override
  _MovieDetailsPageState createState() => _MovieDetailsPageState();
}

class _MovieDetailsPageState extends State<MovieDetailsPage> with SingleTickerProviderStateMixin {
  late Future<Movie?> _movieFuture;
  final MoviedetailsService _apiService = MoviedetailsService();
  final _reviewController = TextEditingController();
  final FirebaseFirestore _firestore = FirebaseFirestore.instance;
  late TabController _tabController;
  @override
  void initState() {
    super.initState();
    _movieFuture = _apiService.getMovieDetails(widget.movieId);
    _tabController = TabController(length: 5, vsync: this);
  }
  @override
  void dispose() {
    _tabController.dispose();
    super.dispose();
  }
  Widget _buildRatingBar(double rating, {int maxRating = 5}) {
    return Row(
      mainAxisAlignment: MainAxisAlignment.min,
      children: [
        Row(
          children: List.generate(maxRating, (index) {
            double fillPercentage = (rating - index) > 1 ? 1 : (rating - index) < 0 ? 0 : (rating - index);
            return Container(
```

```

        width: 12,
        height: 8,
        margin: const EdgeInsets.symmetric(horizontal: 1),
        decoration: BoxDecoration(
          color: Colors.greenAccent.withOpacity(fillPercentage),
          borderRadius: BorderRadius.circular(1),
        ),
      );
    }},
  ),
  const SizedBox(width: 8),
  Text(
    rating.toStringAsFixed(1),
    style: const TextStyle(
      color: Colors.white,
      fontWeight: FontWeight.bold,
    ),
  ),
),
],
);
}

```

```

@override
Widget build(BuildContext context) {
  return Scaffold(
    backgroundColor: Colors.grey[900],
    body: FutureBuilder<Movie?>(
      future: _movieFuture,
      builder: (context, snapshot) {
        if (snapshot.connectionState == ConnectionState.waiting) {
          return const Center(child: CircularProgressIndicator());
        } else if (snapshot.hasError) {
          return Center(child: Text('Error: ${snapshot.error}', style: const TextStyle(color: Colors.white)));
        } else if (snapshot.hasData && snapshot.data != null) {
          final movie = snapshot.data!;
          return CustomScrollView(
            slivers: [
              // App Bar with Movie Poster
              SliverAppBar(
                expandedHeight: 220.0,
                floating: false,
                pinned: true,
                backgroundColor: Colors.grey[950],
                flexibleSpace: FlexibleSpaceBar(
                  title: Text(
                    movie.primaryTitle,
                    style: const TextStyle(
                      color: Colors.white,
                      fontWeight: FontWeight.bold,
                      fontSize: 16,
                    ),
                  ),
                ),
              ),
              background: Stack(
                fit: StackFit.expand,
                children: [
                  movie.primaryImage.isNotEmpty
                    ? CachedNetworkImage(
                        imageUrl: movie.primaryImage,
                        fit: BoxFit.cover,
                        placeholder: (context, url) => const Center(child: CircularProgressIndicator()),
                        errorWidget: (context, url, error) => const Icon(Icons.error),

```

```

    )
    : Container(color: Colors.grey.shade800),
    // Gradient overlay for better text visibility
    const DecoratedBox(
      decoration: BoxDecoration(
        gradient: LinearGradient(
          begin: Alignment.topCenter,
          end: Alignment.bottomCenter,
          colors: [Colors.transparent, Colors.black87],
        ),
      ),
    ),
  ],
),
),
),

// Movie Info
SliverToBoxAdapter(
  child: Padding(
    padding: const EdgeInsets.all(16.0),
    child: Column(
      crossAxisAlignment: CrossAxisAlignment.start,
      children: [
        // Director info
        Text(
          'DIRECTED BY',
          style: TextStyle(
            color: Colors.grey.shade400,
            fontSize: 12,
          ),
        ),
        Text(
          movie.directors.isNotEmpty ? movie.directors.map((d) => d.fullName).join(", ") : 'Unknown',
          style: const TextStyle(
            color: Colors.white,
            fontSize: 14,
            fontWeight: FontWeight.bold,
          ),
        ),
      ],
    ),
  ),

  const SizedBox(height: 12),

  // Year, runtime
  Row(
    children: [
      Text(
        '${movie.startYear}',
        style: const TextStyle(color: Colors.white),
      ),
      const SizedBox(width: 16),
      Text(
        '${movie.runtimeMinutes} mins',
        style: const TextStyle(color: Colors.white),
      ),
      const Spacer(),
      TextButton(
        onPressed: () {},
        child: const Text('TRAILER', style: TextStyle(color: Colors.white)),
      ),
    ],
  ),

```

```

),

const SizedBox(height: 16),

// Movie description
Text(
  movie.description,
  style: TextStyle(
    color: Colors.grey.shade300,
    fontSize: 14,
  ),
),

const SizedBox(height: 24),

// Remove ads button
Center(
  child: OutlinedButton(
    onPressed: () {},
    style: OutlinedButton.styleFrom(
      side: BorderSide(color: Colors.grey.shade600),
      shape: RoundedRectangleBorder(
        borderRadius: BorderRadius.circular(4),
      ),
    ),
    child: Text(
      'REMOVE ADS',
      style: TextStyle(color: Colors.grey.shade400),
    ),
  ),
),

const SizedBox(height: 24),

// Ratings
const Text(
  'RATINGS',
  style: TextStyle(
    color: Colors.white,
    fontSize: 14,
    fontWeight: FontWeight.bold,
  ),
),

const SizedBox(height: 8),

Row(
  children: [
    Expanded(
      child: Container(
        height: 40,
        decoration: BoxDecoration(
          color: Colors.grey.shade900,
          borderRadius: BorderRadius.circular(4),
        ),
      ),
      child: Row(
        mainAxisAlignment: MainAxisAlignment.spaceEvenly,
        children: List.generate(5, (index) {
          return Container(
            width: 6,
            height: 20,

```

```

        color: Colors.grey.shade700,
      );
    }},
  ),
),
),
const SizedBox(width: 12),
_buildRatingBar(movie.averageRating / 2), // Converting to 5-star scale
],
),

const SizedBox(height: 16),

// Rating & Review Button
OutlinedButton.icon(
  onPressed: () {},
  icon: const Icon(Icons.star_border, color: Colors.white),
  label: const Text('Rate, log, review, add to list + more', style: TextStyle(color: Colors.white)),
  style: OutlinedButton.styleFrom(
    side: const BorderSide(color: Colors.white),
    minimumSize: const Size(double.infinity, 40),
  ),
),

const SizedBox(height: 24),

// Where to watch
const Text(
  'Where to watch',
  style: TextStyle(
    color: Colors.white,
    fontSize: 14,
    fontWeight: FontWeight.bold,
  ),
),

const SizedBox(height: 12),

// Streaming services
Row(
  children: [
    Container(
      padding: const EdgeInsets.all(8),
      decoration: BoxDecoration(
        color: Colors.blue,
        borderRadius: BorderRadius.circular(4),
      ),
      child: Image.network(
        'https://api.dicebear.com/7.x/initials/svg?seed=PV',
        width: 20,
        height: 20,
        color: Colors.white,
      ),
    ),
    const SizedBox(width: 8),
    Container(
      padding: const EdgeInsets.all(8),
      decoration: BoxDecoration(
        color: Colors.blue.shade900,
        borderRadius: BorderRadius.circular(4),
      ),
    ),
  ],
)

```

```

        child: Image.network(
          'https://api.dicebear.com/7.x/initials/svg?seed=MAX',
          width: 20,
          height: 20,
          color: Colors.white,
        ),
      ),
    ],
  ),

  const SizedBox(height: 24),

  // Members, Reviews, Lists buttons
  Row(
    children: [
      Expanded(
        child: Container(
          height: 60,
          decoration: BoxDecoration(
            color: Colors.green,
            borderRadius: BorderRadius.circular(4),
          ),
        ),
        child: const Column(
          mainAxisAlignment: MainAxisAlignment.center,
          children: [
            Icon(Icons.person, color: Colors.white),
            Text('Members', style: TextStyle(color: Colors.white, fontSize: 12)),
            Text('463k', style: TextStyle(color: Colors.white, fontSize: 12)),
          ],
        ),
      ),
      const SizedBox(width: 8),
      Expanded(
        child: Container(
          height: 60,
          decoration: BoxDecoration(
            color: Colors.grey.shade800,
            borderRadius: BorderRadius.circular(4),
          ),
        ),
        child: const Column(
          mainAxisAlignment: MainAxisAlignment.center,
          children: [
            Icon(Icons.comment, color: Colors.white),
            Text('Reviews', style: TextStyle(color: Colors.white, fontSize: 12)),
            Text('2.8k', style: TextStyle(color: Colors.white, fontSize: 12)),
          ],
        ),
      ),
      const SizedBox(width: 8),
      Expanded(
        child: Container(
          height: 60,
          decoration: BoxDecoration(
            color: Colors.blue,
            borderRadius: BorderRadius.circular(4),
          ),
        ),
        child: const Column(
          mainAxisAlignment: MainAxisAlignment.center,
          children: [

```



```

        Icon(Icons.list, color: Colors.white),
        Text('Lists', style: TextStyle(color: Colors.white, fontSize: 12)),
        Text('804', style: TextStyle(color: Colors.white, fontSize: 12)),
      ],
    ),
  ),
],
),
const SizedBox(height: 24),

// Related news
Padding(
  padding: const EdgeInsets.only(bottom: 8.0),
  child: Text(
    'RELATED NEWS (1 OF 2)',
    style: TextStyle(color: Colors.grey.shade400, fontSize: 12),
  ),
),

// News card
Container(
  decoration: BoxDecoration(
    color: Colors.grey.shade900,
    borderRadius: BorderRadius.circular(8),
  ),
  child: Row(
    children: [
      ClipRRect(
        borderRadius: const BorderRadius.only(
          topLeft: Radius.circular(8),
          bottomLeft: Radius.circular(8),
        ),
        child: Image.network(
          'https://api.dicebear.com/7.x/shapes/svg?seed=news',
          width: 80,
          height: 80,
        ),
      ),
      const SizedBox(width: 12),
      Expanded(
        child: Column(
          crossAxisAlignment: CrossAxisAlignment.start,
          children: [
            const Text(
              'When Jack Met Letterboxd',
              style: TextStyle(
                color: Colors.white,
                fontWeight: FontWeight.bold,
              ),
            ),
            const SizedBox(height: 4),
            Text(
              'Four favorites with the director and cast of Companion.',
              style: TextStyle(color: Colors.grey.shade400, fontSize: 12),
            ),
          ],
        ),
      ),
    ],
  ),
),
],

```

```

    ),
  ),
],
),
),
),

// Tabs
SliverPersistentHeader(
  delegate: _SliverAppBarDelegate(
    TabBar(
      controller: _tabController,
      labelColor: Colors.white,
      unselectedLabelColor: Colors.grey,
      indicatorColor: Colors.white,
      tabs: const [
        Tab(text: 'CAST + CREW'),
        Tab(text: 'DETAILS'),
        Tab(text: 'GENRE'),
        Tab(text: 'RELEASES'),
        Tab(text: ''),
      ],
    ),
  ),
  pinned: true,
),

// Tab Content - Cast
SliverToBoxAdapter(
  child: SizedBox(
    height: 150,
    child: ListView.builder(
      scrollDirection: Axis.horizontal,
      itemCount: movie.cast.length,
      itemBuilder: (context, index) {
        final actor = movie.cast[index];
        return Container(
          width: 80,
          margin: const EdgeInsets.all(8),
          child: Column(
            children: [
              CircleAvatar(
                radius: 30,
                backgroundColor: Colors.grey.shade800,
                child: Text(
                  actor.fullName.isNotEmpty ? actor.fullName[0] : '?',
                  style: const TextStyle(color: Colors.white),
                ),
              ),
              const SizedBox(height: 4),
              Text(
                actor.fullName,
                style: const TextStyle(color: Colors.white, fontSize: 12),
                maxLines: 2,
                textAlign: TextAlign.center,
                overflow: TextOverflow.ellipsis,
              ),
              Text(
                actor.characters?.join(', ') ?? 'Role',
                style: TextStyle(color: Colors.grey.shade400, fontSize: 10),
                maxLines: 1,

```

```

        overflow: TextOverflow.ellipsis,
      ),
    ],
  ),
);
},
),
),
),

// Crew Section
SliverToBoxAdapter(
  child: Padding(
    padding: const EdgeInsets.only(left: 16.0),
    child: Column(
      crossAxisAlignment: CrossAxisAlignment.start,
      children: [
        const Padding(
          padding: EdgeInsets.symmetric(vertical: 8.0),
          child: Text(
            'CREW',
            style: TextStyle(
              color: Colors.white,
              fontWeight: FontWeight.bold,
            ),
          ),
        ),
      ],
    ),
  ),
  SizedBox(
    height: 100,
    child: ListView.builder(
      scrollDirection: Axis.horizontal,
      itemCount: movie.directors.length,
      itemBuilder: (context, index) {
        final director = movie.directors[index];
        return Container(
          width: 80,
          margin: const EdgeInsets.all(8),
          child: Column(
            children: [
              CircleAvatar(
                radius: 25,
                backgroundColor: Colors.redAccent,
                child: Text(
                  director.fullName.isNotEmpty ? director.fullName[0] : '?',
                  style: const TextStyle(color: Colors.white),
                ),
              ),
              const SizedBox(height: 4),
              Text(
                director.fullName,
                style: const TextStyle(color: Colors.white, fontSize: 12),
                maxLines: 2,
                textAlign: TextAlign.center,
                overflow: TextOverflow.ellipsis,
              ),
              Text(
                'Director',
                style: TextStyle(color: Colors.grey.shade400, fontSize: 10),
                maxLines: 1,
                overflow: TextOverflow.ellipsis,
              ),
            ],
          ),
        );
      },
    ),
  ),
);

```

```

    ],
  ),
);
},
),
),
],
),
),
),
),

// Reviews Section
SliverToBoxAdapter(
  child: Padding(
    padding: const EdgeInsets.all(16.0),
    child: Column(
      crossAxisAlignment: CrossAxisAlignment.start,
      children: [
        const Padding(
          padding: EdgeInsets.symmetric(vertical: 8.0),
          child: Text(
            'REVIEWS',
            style: TextStyle(
              color: Colors.white,
              fontWeight: FontWeight.bold,
            ),
          ),
        ),
      ],
    ),
  ),

// Reviews from Firebase
StreamBuilder<QuerySnapshot>(
  stream: _firestore.collection('reviews').where('movieId', isEqualTo: widget.movieId).snapshots(),
  builder: (context, snapshot) {
    if (snapshot.hasError) {
      return Text('Error: ${snapshot.error}', style: const TextStyle(color: Colors.white));
    }

    if (snapshot.connectionState == ConnectionState.waiting) {
      return const Center(child: CircularProgressIndicator());
    }

    final reviews = snapshot.data?.docs ?? [];

    if (reviews.isEmpty) {
      return const Text(
        'No reviews yet. Be the first to review!',
        style: TextStyle(color: Colors.grey),
      );
    }

    return Column(
      children: reviews.map((doc) {
        final data = doc.data() as Map<String, dynamic>;
        return Container(
          margin: const EdgeInsets.symmetric(vertical: 8),
          child: Row(
            crossAxisAlignment: CrossAxisAlignment.start,
            children: [
              CircleAvatar(
                backgroundColor: Colors.grey.shade800,
                child: Text(

```

```

        (data['user'] as String).isEmpty ? (data['user'] as String)[0] : '?',
        style: const TextStyle(color: Colors.white),
      ),
    ),
    const SizedBox(width: 12),
    Expanded(
      child: Column(
        crossAxisAlignment: CrossAxisAlignment.start,
        children: [
          Text(
            data['user'] ?? 'Anonymous',
            style: const TextStyle(
              color: Colors.white,
              fontWeight: FontWeight.bold,
            ),
          ),
          const SizedBox(height: 4),
          Text(
            data['text'] ?? "",
            style: TextStyle(color: Colors.grey.shade300),
          ),
        ],
      ),
    ),
    _buildRatingBar(4.5),
  ],
),
);
}).toList(),
);
},
),

const SizedBox(height: 16),

// Add Review
TextField(
  controller: _reviewController,
  style: const TextStyle(color: Colors.white),
  decoration: InputDecoration(
    labelText: 'Add your review',
    labelStyle: TextStyle(color: Colors.grey.shade400),
    enabledBorder: OutlineInputBorder(
      borderSide: BorderSide(color: Colors.grey.shade700),
    ),
    focusedBorder: const OutlineInputBorder(
      borderSide: BorderSide(color: Colors.white),
    ),
  ),
),
maxLines: 3,
),
const SizedBox(height: 8),
ElevatedButton(
  onPressed: () {
    _addReview(movie.id);
  },
  style: ElevatedButton.styleFrom(
    backgroundColor: Colors.green,
    minimumSize: const Size(double.infinity, 40),
  ),
  child: const Text('SUBMIT REVIEW'),

```

```

    ),

    const SizedBox(height: 16),

    // View All Reviews
    Center(
      child: TextButton(
        onPressed: () {},
        child: const Text(
          'ALL REVIEWS',
          style: TextStyle(color: Colors.white),
        ),
      ),
    ),
  ],
),
),
],
);
} else {
  return const Center(
    child: Text(
      'Movie not found',
      style: TextStyle(color: Colors.white),
    ),
  );
}
},
),
);
}

Future<void> _addReview(String movieId) async {
  if (_reviewController.text.isNotEmpty) {
    await _firestore.collection('reviews').add({
      'movieId': movieId,
      'text': _reviewController.text,
      'user': 'Anonymous', // Replace with actual user authentication
      'timestamp': FieldValue.serverTimestamp(),
      'rating': 4.5, // This would ideally come from a rating widget
    });
    _reviewController.clear();
    setState(() {}); // Refresh the UI
  }
}

// Helper class for the persistent header
class _SliverAppBarDelegate extends SliverPersistentHeaderDelegate {
  final TabBar _tabBar;

  _SliverAppBarDelegate(this._tabBar);

  @override
  double get minExtent => _tabBar.preferredSize.height;

  @override
  double get maxExtent => _tabBar.preferredSize.height;

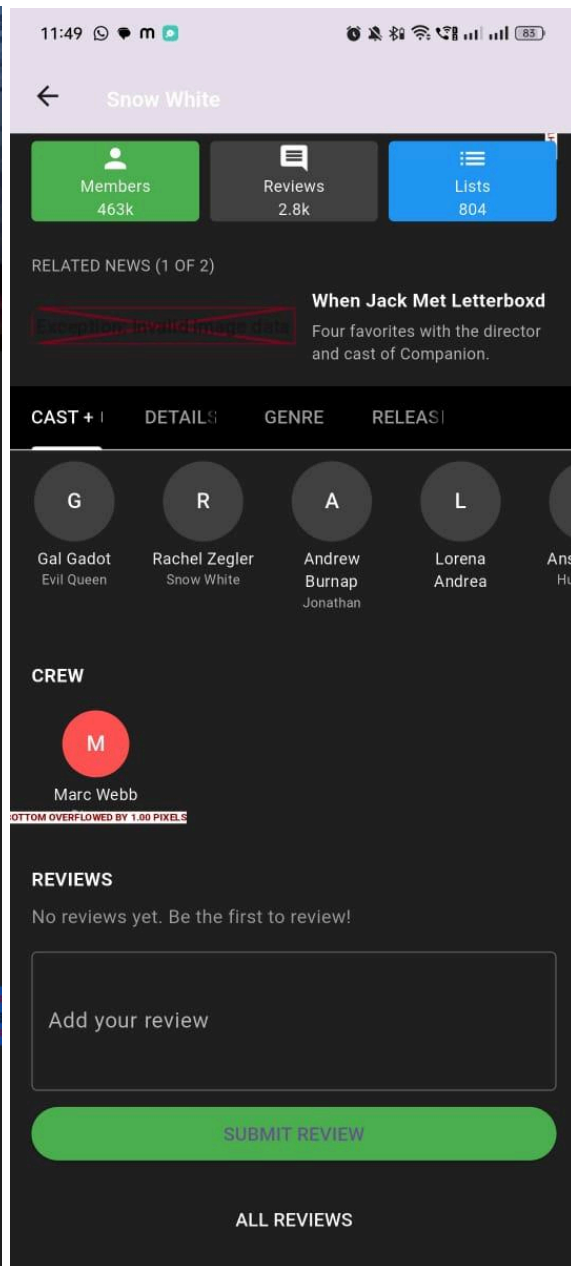
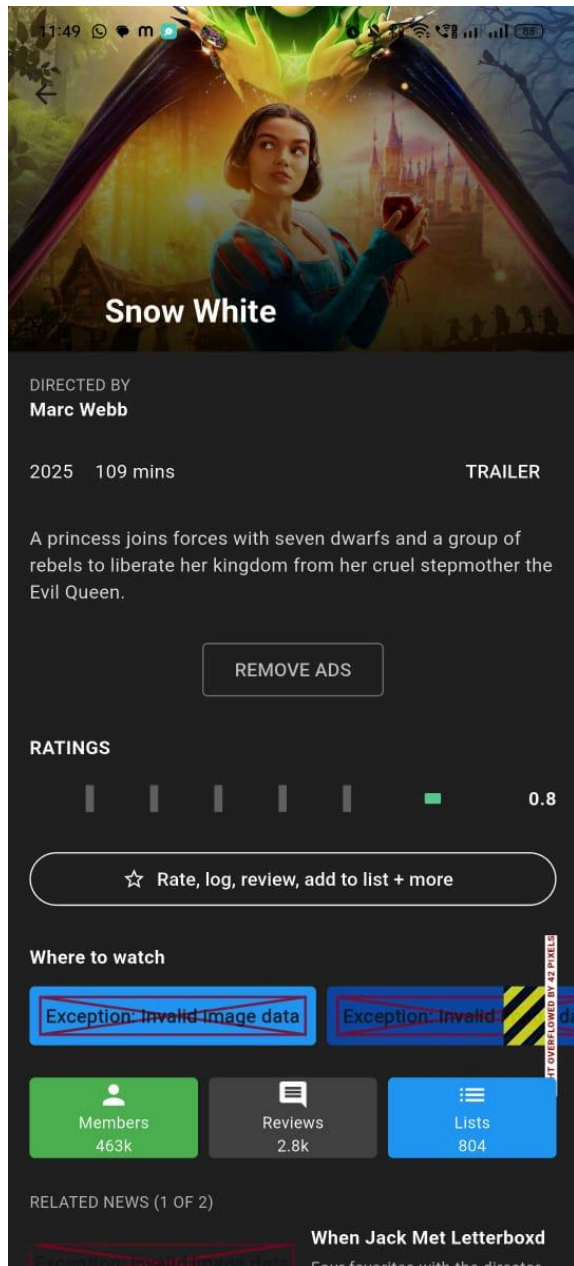
  @override

```

```
Widget build(BuildContext context, double shrinkOffset, bool overlapsContent) {
  return Container(
    color: Colors.black,
    child: _tabBar,
  );
}

@override
bool shouldRebuild(_SliverAppBarDelegate oldDelegate) {
  return false;
}
}
```

Screenshots:



Conclusion:

In conclusion, incorporating icons, images, and fonts into a Flutter app significantly enhances the user interface and user experience. By correctly managing assets through the pubspec.yaml file and understanding how to use built-in and custom resources, developers can build visually appealing and functional applications. Whether using asset images, network images, Material icons, SVGs, or custom fonts, Flutter provides flexible and efficient ways to integrate these elements. Mastery of these features empowers developers to design clean, modern, and engaging mobile applications that align with branding and usability standards.