EXPERIMENT 3

NAME-SPANDAN DEB

CLASS-D15A

ROLL NO-13

AIM-To include images, fonts in flutter app.

THEORY-

Images are an essential part of UI design, and Flutter supports adding both local and network images.

A) Local images can be stored in the project directory and loaded into the app.

Steps to add Local images

- Create an assets folder in the root directory.
- Store images inside the assets folder.
- Declare assets in pubspec.yaml under the flutter section:
  flutter:
    assets:
      - assets/image1.png
      - assets/images/image2.jpg

B) Network Images

Flutter allows displaying images from the internet using Image.network():

Image.network('https://example.com/image.jpg')

Font Awesome provides a vast collection of scalable vector icons that behave like fonts. These icons can be used in Flutter via the font_awesome_flutter package, which integrates Font Awesome's font-based icons seamlessly into the app.

SYNTAX

1)Create an assets folder for Local images.

Declare assets in pubspec.yaml file.

flutter:

 assets:

   - assets/image1.png

   - assets/images/image2.jpg

Image.asset('assets/image1.png')

2)If using network Images

Image.network('https://example.com/image.jpg')

3)Install fontawesome package in flutter

Add this dependency in pubspec.yaml file

dependencies:

  font_awesome_flutter: ^10.7.0

Run flutter pub get

FaIcon(FontAwesomeIcons.heart, size: 50, color: Colors.red)

Widget properties

1)image

- width: Sets image width.
- height: Sets image height.
- fit: Controls how image fits (e.g., BoxFit.cover, BoxFit.fill).
- alignment: Aligns the image inside the container.
- color: Applies a color filter.
- opacity: Controls image transparency.
- loadingBuilder: Handles loading states.
- errorBuilder: Handles image load errors.

Example

```
Image.network(
  'https://example.com/image.jpg',
  width: 100,
  height: 100,
  fit: BoxFit.contain,
  loadingBuilder: (context, child, progress) {
    return progress == null ? child : CircularProgressIndicator();
  },
  errorBuilder: (context, error, stackTrace) {
    return Icon(Icons.error);
  },
)
```

2)font

- size: Adjusts icon size.
- color: Sets icon color.
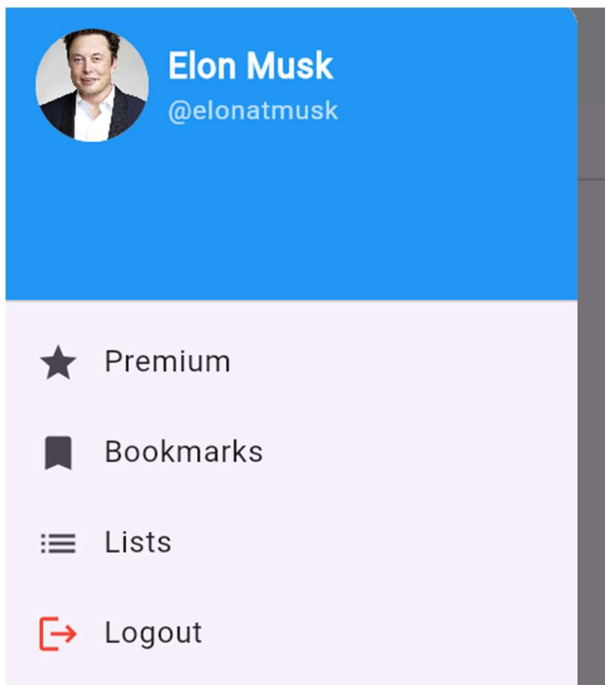- semanticLabel: Adds an accessibility label for screen readers.

Example:
```
FaIcon(
  FontAwesomeIcons.heart,
  size: 50,        // Sets icon size
  color: Colors.red, // Sets icon color
  semanticLabel: 'Heart Icon', // Provides accessibility label
)
```

CODE

ListTile(leading: Icon(Icons.star), title: Text("Premium"), onTap: () {}),

ListTile(leading: Icon(Icons.bookmark), title: Text("Bookmarks"), onTap: () {}),

ListTile(leading: Icon(Icons.list), title: Text("Lists"), onTap: () {}),

ListTile(

leading: Icon(Icons.logout, color: Colors.red), title: Text("Logout"), onTap: _logout),

],

OUTPUT

CODE

```
child: ListView.builder(
        itemCount: tweets.length,
        itemBuilder: (context, index) {
          final tweet = tweets[index];
          return Card(
            child: ListTile(
              leading: CircleAvatar(backgroundImage:
NetworkImage(tweet["userAvatar"])),
                title: Text(tweet["username"]),
                subtitle: Column(
                  crossAxisAlignment: CrossAxisAlignment.start,
                  children: [
                    Text(tweet["content"]),
                    if (tweet["image"] != null) Image.memory(tweet["image"],
height: 100),
                    Row(
                      children: [
                        IconButton(
                            icon: Icon(Icons.thumb_up, color: Colors.blue),
                            onPressed: () => _likeTweet(index)),
                        Text("${tweet["likes"]}"),
                        IconButton(icon: Icon(Icons.comment), onPressed: () {}),
                        IconButton(icon: Icon(Icons.share), onPressed: () {}),
                        IconButton(
                            icon: Icon(Icons.delete, color: Colors.red),
```

```
                        onPressed: () => _deleteTweet(index)),
                ],
              ),
          ],
```

OUTPUT