MAD PWA LAB 3

AIM:

To include icons, images, fonts in flutter app.

THEORY:

1. Icons:

Flutter provides the Icon widget for displaying material design icons. We can also use icon packages like font_awesome_flutter for additional icon options.

2.lmages:

We can use the Image widget to display images in your Flutter app. Images can be loaded from assets, the internet, or other sources.

3.Fonts:

To use custom fonts in Flutter, you need to include the font files (usually .ttf or .otf) in your project and then reference them in your Flutter code.

SYNTAX:

2.lmages:

```
void _getImageAndUpload() async {
  final pickedFile = await _picker.pickImage(source: ImageSource.gallery);
  if (pickedFile != null) {
    // Valid file path obtained
    print('File Path: ${pickedFile.path}');
```

```
// Check if the file exists
   File imageFile = File(pickedFile.path);
   if (await imageFile.exists()) {
     setState(() {
      _imageFile = imageFile;
     });
     String? downloadURL = await DatabaseMethods().uploadImage(_imageFile!);
     if (downloadURL != null) {
      // Image uploaded successfully
      print('Image uploaded. Download URL: $downloadURL');
      // Update the UI with the new story URL and timestamp
      setState(() {
       _stories.add(downloadURL);
       _storyTimestamps.add(DateTime.now());
      });
     } else {
      // Image upload failed
      print('Image upload failed.');
     }
   } else {
     // File does not exist
     print('Selected file does not exist.');
   }
  } else {
   // File picker returned null
   print('No file selected.');
3.Fonts:
// Loading custom font
Text(
'Custom Font Text',
style: TextStyle(
fontFamily: 'Font Family',
fontSize: 16,
```

}

),);

WIDGETS AND PROPERTIES:

The widgets we focussed in this lab are:

- Images- We added the X logo image on the page.
- Icons- We added a facebook icon on the page from the FontAwesomeIcons library along with person and lock icons for username and password field.
- Fonts

CODE:

```
import 'dart:io';
import 'package:flutter/material.dart';
import 'package:image_picker/image_picker.dart';
import 'package:messenger_clone/services/database.dart';
import 'dart:convert';
class StoriesScreen extends StatefulWidget {
 @override
 _StoriesScreenState createState() => _StoriesScreenState();
}
class _StoriesScreenState extends State<StoriesScreen> {
 List<String> _stories = [];
 List<DateTime> _storyTimestamps = [];
```

```
File? _imageFile;
final ImagePicker _picker = ImagePicker();
@override
void initState() {
 super.initState();
 _loadStories();
}
void _loadStories() async {
 // Implement logic to fetch stories from Firestore
 // For now, I'm just setting some placeholder URLs and timestamps
 setState(() {
  _stories = [
   'https://example.com/image1.jpg',
   'https://example.com/image2.jpg',
   // Add more story URLs here
  ];
  _storyTimestamps = [
   DateTime.now().subtract(Duration(hours: 3)), // Example timestamp
   DateTime.now().subtract(Duration(minutes: 30)), // Example timestamp
```

```
// Add more timestamps here
  ];
 });
}
void _getImageAndUpload() async {
 final pickedFile = await _picker.pickImage(source: ImageSource.gallery);
 if (pickedFile != null) {
  // Valid file path obtained
  print('File Path: ${pickedFile.path}');
  // Check if the file exists
  File imageFile = File(pickedFile.path);
  if (await imageFile.exists()) {
   setState(() {
    _imageFile = imageFile;
   });
   String? downloadURL = await DatabaseMethods().uploadImage(_imageFile!);
   if (downloadURL != null) {
```

```
// Image uploaded successfully
   print('Image uploaded. Download URL: $downloadURL');
   // Update the UI with the new story URL and timestamp
   setState(() {
    _stories.add(downloadURL);
    _storyTimestamps.add(DateTime.now());
   });
  } else {
   // Image upload failed
   print('Image upload failed.');
  }
 } else {
  // File does not exist
  print('Selected file does not exist.');
 }
} else {
 // File picker returned null
 print('No file selected.');
}
```

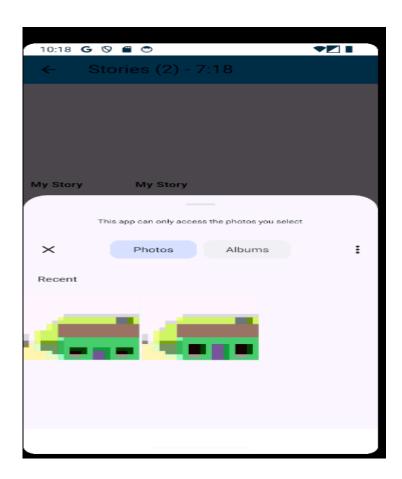
}

```
@override
 Widget build(BuildContext context) {
  return Scaffold(
   appBar: AppBar(
    title: Text(_storyTimestamps.isNotEmpty
       ? 'Stories (${_storyTimestamps.length}) -
${_storyTimestamps.first.hour}:${_storyTimestamps.first.minute}'
       : 'Stories'),
   ),
   body: _stories.isEmpty
     ? Center(
        child: Text('No stories available'),
      )
      : ListView.builder(
        itemCount: _stories.length,
        scrollDirection: Axis.horizontal,
        itemBuilder: (context, index) {
         return Padding(
          padding: EdgeInsets.symmetric(horizontal: 8.0),
          child: Column(
           crossAxisAlignment: CrossAxisAlignment.start,
           mainAxisAlignment: MainAxisAlignment.start,
```

```
children: [
 Container(
  width: 100,
  height: 180,
  decoration: BoxDecoration(
   borderRadius: BorderRadius.circular(8.0),
   image: DecorationImage(
    image: NetworkImage(_stories[index]),
    fit: BoxFit.cover,
   ),
  ),
 ),
 SizedBox(height: 8.0),
 // Show username only when a story is available
 Text(
  'My Story', // Display username or other info here
  style: TextStyle(
   fontWeight: FontWeight.bold,
  ),
 ),
],
```

```
),
);
},
),
floatingActionButton: FloatingActionButton(
onPressed: _getImageAndUpload,
child: lcon(lcons.camera_alt),
),
);
}
```

OUTPUT:



CONCLUSION:

- 1. Learnt about fonts, images and icons in flutter.
- 2. Imported a new library for icons and added it to the pubspec.yaml file. 3. Faced errors in adding image due to usage of assets syntax instead of network syntax while using image from browser.