

Experiment 4

AIM: To understand the working of forms, authentication, and storage in the mobile app.

THEORY: Data input, authentication, and storage are fundamental aspects of mobile application development. A well-designed data input system allows users to provide necessary information through forms, while authentication ensures secure access. Once collected, the data needs to be stored efficiently for future retrieval and processing.

1. Authentication in Mobile Apps

User authentication is a critical component of mobile applications, ensuring secure access to user-specific data. Firebase Authentication provides multiple authentication methods, including:

Email and Password Authentication: Users register using their email and password, which Firebase securely stores.

Google Sign-In: Allows users to log in with their Google accounts for a seamless experience.

Anonymous Authentication: Enables temporary access without requiring user credentials.

2. Data Input in Mobile Apps

Mobile applications require user input for various functionalities, including product addition, user registration, and order processing. The common methods for data input include:

Text Fields: Used for entering names, product details, or any textual information.

Checkboxes: Allow users to select multiple options from a list.

Dropdown Menus (DropDownButton): Useful for selecting one option from a predefined list.

3. Data Storage in Mobile Apps

Once user input is collected, it needs to be stored persistently. Mobile applications offer different storage mechanisms:

1. Local Storage

Stores data on the user's device.

Suitable for offline access and temporary data storage.

Examples: SharedPreferences, Hive (for key-value storage in Flutter).

2. Firebase Firestore Database

A cloud-hosted NoSQL database.

Used for storing structured data like user profiles and product listings.

Provides real-time data updates and seamless synchronization across devices.

3. Firebase Storage

Used for storing images, files, and other media assets.

Allows secure access and retrieval via Firebase authentication.

Experiment Implementation in BlingBay

In this experiment, we have implemented:

1. User Authentication:

- Users can register using Firebase Authentication with an email and password.
- Login functionality enables secure access to user accounts.
- Firebase stores user credentials securely and provides authentication state management.

2. Product Addition Form:

- Collects the following information:
 - Product Name (Text Input)
 - User information
 - Price (Numeric Input Field)

For data storage, we use Firebase Firestore, which allows efficient handling of product information and real-time updates. Image files are uploaded to Firebase Storage, and their URLs are stored in Firestore.

By fetching real-time data from Firebase Firestore, the UI remains updated whenever a new product is added or modified.

code:

```
import 'package:bingbayy/LoginPage.dart';
import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';

class Signup extends StatefulWidget {
  @override
  _SignupState createState() => _SignupState();
}

class _SignupState extends State<Signup> {
  final TextEditingController _emailController =
    TextEditingController();
  final TextEditingController _passwordController =
    TextEditingController();
  final TextEditingController _confirmPasswordController =
    TextEditingController();
  bool _isLoading = false;

  Future<void> _signup() async {
    if (_passwordController.text !=
    _confirmPasswordController.text) {
      ScaffoldMessenger.of(context).showSnackBar(
        SnackBar(content: Text('Passwords do not match')),
      );
      return;
    }

    setState(() => _isLoading = true);

    try {
      MainAxisAlignment.center,
      children: [
        const SizedBox(height: 20),
        Text(
          'Create Account',
          style: TextStyle(
            fontSize: 26,
            fontWeight: FontWeight.bold,
            color: isDarkMode ? Colors.white : Colors.black,
          ),
        ),
        const SizedBox(height: 10),
        Text(
          'Sign up to get started',
          style: TextStyle(fontSize: 16, color: isDarkMode ?
Colors.white70 : Colors.black54),
```

```
UserCredential userCredential = await
FirebaseAuth.instance.createUserWithEmailAndPassword(
  email: _emailController.text.trim(),
  password: _passwordController.text.trim(),
);

// Save user data to Firestore
await
FirebaseFirestore.instance.collection('users').doc(user
Credential.user!.uid).set({
  'email': _emailController.text.trim(),
  'createdAt': DateTime.now(),
});

// Navigate to Login Screen after successful signup
Navigator.pushReplacement(
  context,
  MaterialPageRoute(builder: (context) =>
LoginScreen()),
);
} on FirebaseAuthException catch (e) {
  ScaffoldMessenger.of(context).showSnackBar(
    SnackBar(content: Text(e.message ?? 'Signup
failed')),
  );
} finally {
  setState(() => _isLoading = false);
}
}

@override
Widget build(BuildContext context) {
  bool isDarkMode = Theme.of(context).brightness ==
Brightness.dark;

  return Scaffold(
    Icon(Icons.lock, color: isDarkMode ? Colors.white70 :
Colors.black54),
    hintText: 'Password',
    filled: true,
    fillColor: isDarkMode ?
Colors.grey.shade900 : Colors.grey.shade200,
    border: OutlineInputBorder(borderRadius:
BorderRadius.circular(12), borderSide:
BorderSide.none),
  ),
  style: TextStyle(color: isDarkMode ?
```

```

    ),
    const SizedBox(height: 30),

    /// Email Field
    TextField(
      controller: _emailController,
      decoration: InputDecoration(
        prefixIcon: Icon(Icons.email, color: isDarkMode
? Colors.white70 : Colors.black54),
        hintText: 'Email',
        filled: true,
        fillColor: isDarkMode ? Colors.grey.shade900 :
Colors.grey.shade200,
        border: OutlineInputBorder(borderRadius:
BorderRadius.circular(12), borderSide: BorderSide.none),
      ),
      style: TextStyle(color: isDarkMode ? Colors.white
: Colors.black),
    ),
    const SizedBox(height: 15),

    /// Password Field
    TextField(
      controller: _passwordController,
      obscureText: true,
      decoration: InputDecoration(
        prefixIcon: MainAxisAlignment.center,
        children: [
          const SizedBox(height: 20),
          Text(
            'Create Account',
            style: TextStyle(
              fontSize: 26,
              fontWeight: FontWeight.bold,
              color: isDarkMode ? Colors.white : Colors.black,
            ),
          ),
        ],
      ),
      style: TextStyle(fontSize: 16, color: isDarkMode ?
Colors.white70 : Colors.black54),
    ),
    const SizedBox(height: 30),

    /// Email Field
    TextField(
      controller: _emailController,
      decoration: InputDecoration(
        prefixIcon: Icon(Icons.email, color: isDarkMode
? Colors.white70 : Colors.black54),
        hintText: 'Email',
        filled: true,
        fillColor: isDarkMode ? Colors.grey.shade900 :
Colors.grey.shade200,
        border: OutlineInputBorder(borderRadius:
BorderRadius.circular(12), borderSide: BorderSide.none),
      ),
      style: TextStyle(color: isDarkMode ? Colors.white
: Colors.black),
    ),

```

```

Colors.white : Colors.black),
    ),
    const SizedBox(height: 15),

    /// Confirm Password Field
    TextField(
      controller: _confirmPasswordController,
      obscureText: true,
      decoration: InputDecoration(
        prefixIcon: Icon(Icons.lock, color:
isDarkMode ? Colors.white70 : Colors.black54),
        hintText: 'Confirm Password',
        filled: true,
        fillColor: isDarkMode ?
Colors.grey.shade900 : Colors.grey.shade200,
        border: OutlineInputBorder(borderRadius:
BorderRadius.circular(12), borderSide:
BorderSide.none),
      ),
      style: TextStyle(color: isDarkMode ?
Colors.white : Colors.black),
    ),
    const SizedBox(height: 20),

    /// Signup Button
    SizedBox(
      width: double.infinity,
      child: ElevatedButton(
        onPressed: _isLoading ? null : _signup,
        style: ElevatedButton.styleFrom(
          backgroundColor: Colors.blue.shade900,
          padding: const
SizedBox(height: 20),
        ),
      ),

    /// Navigate to Login
    Row(
      mainAxisAlignment:
MainAxisAlignment.center,
      children: [
        Text("Already have an account?", style:
TextStyle(color: isDarkMode ? Colors.white70 :
Colors.black54)),
        TextButton(
          onPressed: () =>
Navigator.pushReplacement(
          context,
          MaterialPageRoute(builder: (context) =>
LoginScreen()),
        ),
        child: const Text('Login', style:
TextStyle(color: Colors.blue)),
      ),
    ],
  ),
);
}
}

```

OUTPUT:

BlingBag

Create account

Aiman

aiman901@gmail.com

Password must be at least 6 character

Create Account

Already a customer

Sign In

User already exists

BlingBag

Create account

Aiman

aiman901@gmail.com

Enter valid password

Enter valid password

Password must be at least 6 character

Create Account

Already a customer

Sign In

BlingBag

Create account

Aiman

aiman901@gmail.com

Password must be at least 6 character

Create Account

Already a customer

Sign In

BlingBag

Sign in with your email and password

Sign In

Forgot Password

aiman901@gmail.com

Show Password

Keep sign in

Sign In

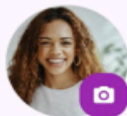
New to Blinbay

Create a new account

11:21 94

11:21 AM 25/05

Edit Profile



Name

Melissa Peters

Email

meipeters@gmail.com

Password

Date of Birth

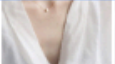
23/05/1995

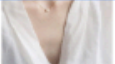
Country/Region


Nigeria


Save

Add Product









necklace

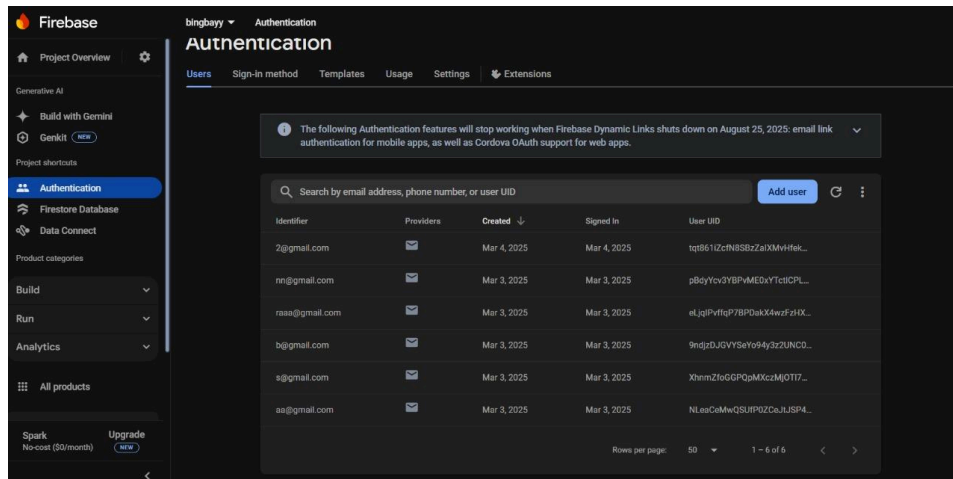
silver necklace

1,234

3

Makeup

Add Product



- stored product details securely using firebase

CONCLUSION:

In BlingBay, we implemented user authentication using Firebase Authentication along with a product entry form. Product details were securely stored using Firebase Firestore and Firebase Storage. The stored information is displayed dynamically in the app's UI, allowing seamless product management and secure access. This experiment helped us understand the importance of authentication, data input, storage, and retrieval in mobile app development, ensuring a smooth and secure user experience.