# Pakistan Ordnance Factories (POF)

## Internship Weekly Report – Week 4

**Name**: Ahmer Moon Majid

**Instructor**: Madam Aansa

**Department**: IT Department

**Technology**: Flutter (Mobile App Development)

**Duration**: 6 Weeks

**Week**: 4

**Report Date**: 9/2/2025

---

# Task: POF e-Services App with Firebase Integration

Assigned Task:

The primary task for this week was to develop a multi-screen clone of the Pakistan Ordnance Factories (POF) e-Services employee portal application and integrate it with Firebase for backend functionality. The objective was to replicate the official app's look, feel, and core navigation flow while implementing user authentication, real-time data persistence, and cloud-based operations using Firebase services such as Firebase Authentication and Cloud Firestore.

---

Key Implementation Details:

1. **Firebase Configuration**

- Integrated Firebase into the Flutter project using the firebase_core package.
- Configured platform-specific setup for Android, iOS, and Web using DefaultFirebaseOptions.
- Initialized Firebase in the main.dart file using Firebase.initializeApp().

2. **Authentication Service (AuthService)**

- Created a dedicated AuthService class to handle all authentication-related operations.
- Implemented signInWithEmailAndPassword for user login.
- Added signOut functionality to log users out securely.
- Integrated registerComplaint method to store complaint data in Firestore.

3. **User Authentication**

- **Login Screen:** Users enter their email and password. Credentials are validated using Firebase Auth.
- **Change Password Screen:** Users can update their password after reauthentication. Implemented using updatePassword and reauthenticateWithCredential methods.

4. **Complaint Management**

- **Register Complaint Screen:** Users can submit complaints with details like category, sub-category, area, bungalow, and description. Data is stored in Firestore.

- **Open & Closed Complaints Screens:** Complaints are fetched in real-time from Firestore using streams and displayed in a structured list view.

5. **Pay Slip Generation (CMA Screen)**
   - Implemented PDF generation for pay slips using the pdf and printing packages.
   - Added platform-specific handling for mobile and web to download pay slips.
6. **State Management & Navigation**
   - Used Provider for state management to handle authentication state across screens.
   - Implemented named routes for seamless navigation between screens.
7. **UI Consistency & Theming**
   - Maintained a consistent green-themed UI across all screens.
   - Used custom background images and semi-transparent overlays for a professional look.
   - Ensured responsive design for both mobile and web platforms.

---

Code Structure Overview:

1. **Firebase Initialization**
   - firebase_options.dart: Contains platform-specific Firebase configuration.
   - main.dart: Initializes Firebase and sets up the app with MultiProvider.
2. **Screens**
   - splash_screen.dart: Displays an animated splash screen before navigating to the login screen.
   - login_screen.dart: Handles user login using Firebase Auth.
   - home_screen.dart: Displays a grid of services and navigates to respective screens.
   - complaint_screen.dart: Shows complaint statistics and actions (Register, Open, Closed).
   - register_complaint_screen.dart: Form to register a new complaint.

- open_complaints_screen.dart & closed_complaints_screen.dart: Display lists of open and closed complaints fetched from Firestore.
- cma_screen.dart: Displays pay slips and allows PDF download.
- change_password_screen.dart: Allows users to change their password after reauthentication.
- login_instructions_screen.dart: Provides login instructions with a mock video player.

3. **Services**

- auth_service.dart: Handles authentication and Firestore operations.

4. **Assets**

- Background images (bg.png), icons, and logos are stored in the assets folder.

---

Firebase Integration Highlights:

1. **Authentication**

- Used FirebaseAuth for user sign-in and sign-out.
- Implemented password update with reauthentication for security.

2. **Cloud Firestore**

- Stored complaint data in Firestore with fields
  like category, area, bungalow, description, and timestamp.
- Used StreamBuilder to listen for real-time updates on open and closed complaints.

3. **Cross-Platform Support**

- Configured Firebase for Android, Web, and Windows platforms.
- Handled platform-specific operations like PDF download using kIsWeb checks.

---

Challenges & Solutions:

1. **Reauthentication for Password Change:**
    - Implemented EmailAuthProvider.credential to reauthenticate users before updating passwords.

2. **Real-Time Data Sync:**
    - Used Stream<QuerySnapshot> to fetch and display complaints in real-time.
3. **Platform-Specific PDF Download:**
    - Used dart:io for mobile and dart:html for web to handle file downloads.
4. **UI Consistency:**
    - Maintained a uniform design across screens using reusable widgets and a global theme.

---

Output:

The application successfully replicates the POF e-Services portal with the following features:

- User authentication via Firebase.
- Real-time complaint registration and tracking.
- Pay slip generation and download.
- Password change functionality.
- Responsive and visually consistent UI.

---

Conclusion:

This week, I successfully integrated Firebase into the POF e-Services app, enabling user authentication, real-time data persistence, and cross-platform functionality. The app now closely mirrors the official portal in both design and functionality, with the added advantage of cloud-based operations.

---

Output Screenshot:

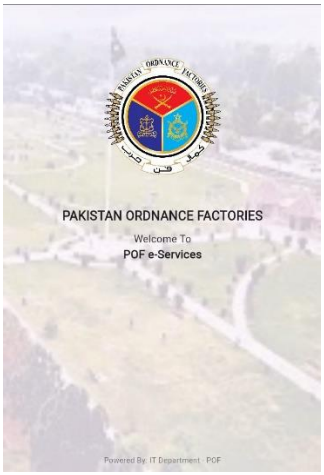


*Figure 2: Splash Screen*

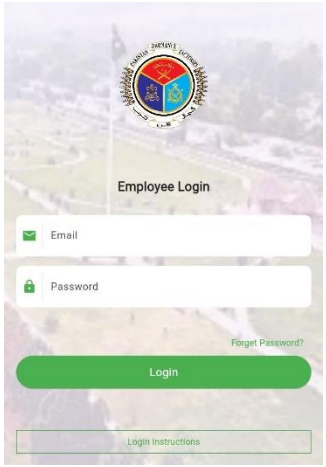


*Figure 1: Login Screen*

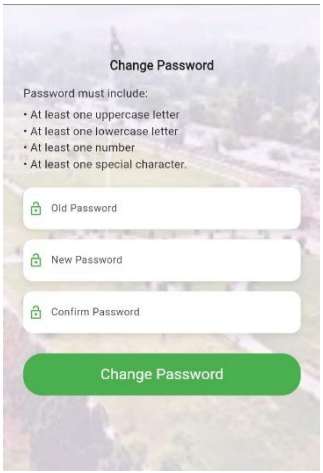


*Figure 4: Change Password Screen*

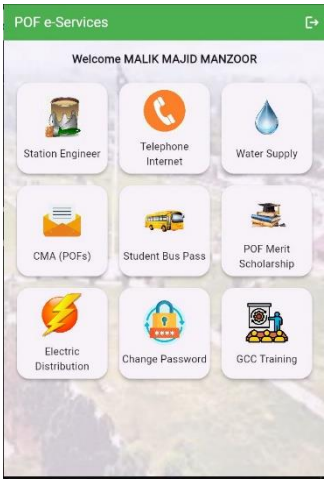


*Figure 3: Home Screen*

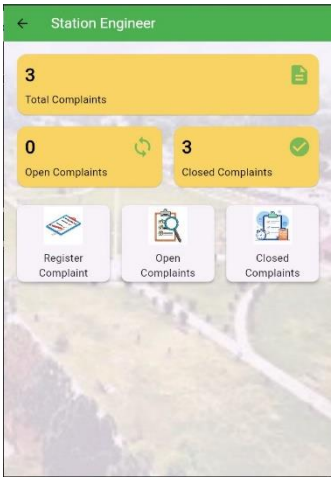


*Figure 5: Complaints Screen*

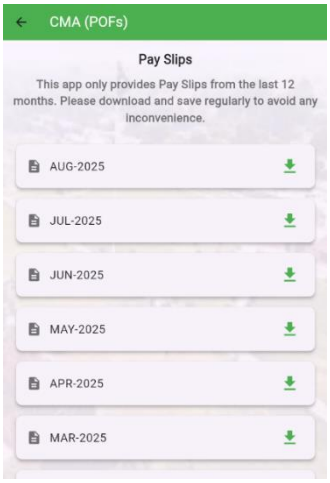


*Figure 6: CMA Screen*

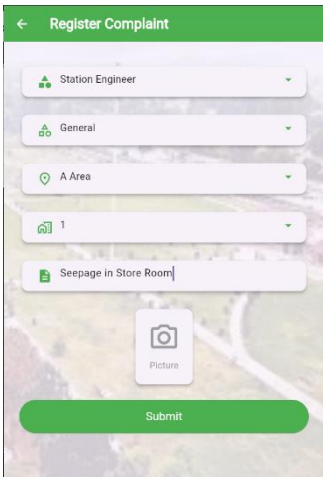


*Figure 9: Register Complaint Screen*

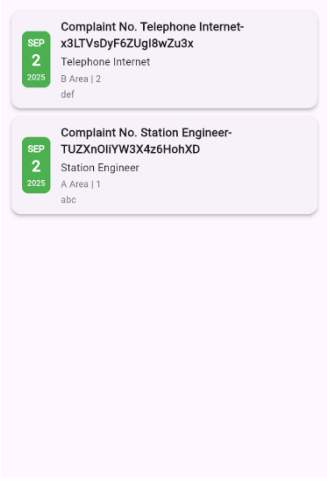


*Figure 8: Opened/Closed Complaints Screen*

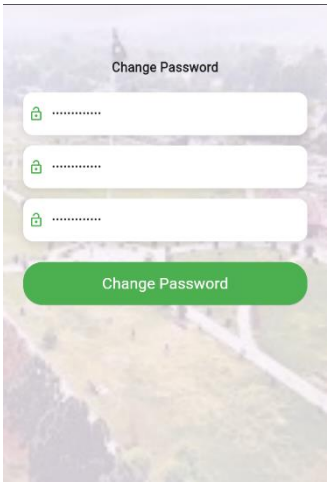


*Figure 7: Change Password Screen*