Festora – Smart Event Hosting and Management App

Course: Android Mobile Development

Instructor: Adin Ashby

Developed by: Aditya Sharma, Harsh Parmar, Nand Kapatel

Institution: LaSalle College, Montreal

# Abstract

Festora is an Android-based mobile application designed to simplify event hosting, discovery, and management. Developed in Java using Android Studio and integrated with Firebase, the app allows users to securely create, view, and manage events. Core features include Firebase Authentication for login, Cloud Firestore for storing event data, and Firebase Storage for event images. Festora aims to provide a real-time, cloud-backed, user-friendly platform for digital event management.

# Introduction

Festora was created to address the growing need for an easy-to-use, mobile-first event management system. Unlike traditional event platforms, Festora offers both hosting and discovery capabilities in one app. By leveraging Firebase’s cloud services, it ensures data persistence, scalability, and security for all user interactions.

# Objectives

• Provide a unified mobile platform for creating and managing events.  
• Ensure secure user authentication through Firebase.  
• Offer real-time event synchronization with Firestore.  
• Deliver a visually appealing Material Design interface.  
• Support image uploads and event categorization.

# Tools and Technologies

• Android Studio (Java)  
• Firebase Authentication  
• Cloud Firestore  
• Firebase Storage  
• Material Design Components  
• XML Layouts & RecyclerView  
• Gradle Build System

# System Architecture

Festora follows a client-cloud architecture where the Android application acts as the front-end client and Firebase provides the back-end services for authentication, storage, and database operations. Key communication flow:  
  
Android App → Firebase Authentication → Firestore Database → Firebase Storage

# App Workflow

1. Launch the app and select Sign In or Sign Up.  
2. Authenticate using Firebase.  
3. View available events on the Home screen.  
4. Create a new event via CreateEventActivity.  
5. Upload event details and images to Firestore.  
6. View or manage hosted events under the Profile section.

# Key Features

• Login & Registration (Firebase Auth)  
• Create & Manage Events  
• Real-time Event Listing  
• Profile Management  
• Notifications & Support Pages  
• Material Design UI

# Code Explanation

• HomeActivity.java – Displays events from Firestore.  
• CreateEventActivity.java – Handles event creation and uploads.  
• EventAdapter.java – Binds event data to RecyclerView.  
• Event.java – Defines event model structure.  
• LoginActivity.java / SignupActivity.java – Manage authentication.  
• ProfileActivity.java – Displays user profile and hosted events.

# Challenges and Solutions

• Firebase Integration: Required configuring google-services.json and authentication setup.  
• Async Operations: Solved by chaining listeners (onSuccess / onFailure).  
• Navigation: Managed smooth activity transitions using explicit intents and flags.

# Testing and Results

The application was tested on multiple Android versions. All core functionalities performed successfully: authentication, event creation, Firestore reads/writes, and image uploads. RecyclerView efficiently displayed events in real-time, and UI navigation was smooth.

# Future Enhancements

• Add ticket booking and event analytics.  
• Implement push notifications.  
• Add real-time chat support.  
• Introduce personalized event recommendations.

# Conclusion

Festora is a fully functional Android application demonstrating the integration of Firebase services with Android front-end development. It showcases real-time data handling, authentication, and storage in a clean, interactive UI. The project successfully meets its goal of simplifying event management and establishes a strong base for future feature expansion.

# Credits

Developed by:  
Aditya Sharma  
Harsh Parmar  
Nand Kapatel  
Instructor: Adin Ashby  
LaSalle College, Montreal