COLLEGE CODE : 9623

COLLEGE NAME : Amrita College of Engineering And Technology

DEPARTMENT : Computer Science and Engineering

STUDENT NM-ID : 6EA79657AE5A46A95504A5BA2B8B3583

ROLL NO : 23CS030

DATE : 11-09-2025

Completed the project named as

Phase 2 Solution Design & Architecture

PROJECT NAME : LOGIN AUTHENTICATION SYSTEM

SUBMITTED BY,

NAME : DEVANANTH RS
MOBILE NO : 7598844814

Phase 2 — Solution Design & Architecture

Login Authentication System

Tech Stack Selection

The technology stack is centered around Firebase, a comprehensive development platform backed by Google.This choice offers a suite of integrated services that streamline development and ensure scalability.

- **Frontend:** The frontend can be developed using any modern JavaScript framework like React, Angular, or Vue.js, or for mobile applications, native Android (Java/Kotlin), or iOS (Swift). These frameworks are well-supported by Firebase SDKs.
- **Backend as a Service (BaaS):** Firebase Authentication will serve as the core of the authentication system. It provides ready-made UI libraries and SDKs to handle user authentication with various providers, including Google.
- **Database:** Cloud Firestore or Realtime Database will be used to store user data beyond the basic profile information provided by the authentication service.
- **Serverless Functions:** Cloud Functions for Firebase can be employed for any necessary backend logic, such as custom user data validation or integration with other services.

UI Structure / API Schema Design

A clean and intuitive user interface is crucial for a seamless user experience. The API design will focus on secure and efficient communication between the client application and Firebase.

- **UI Structure:** For rapid development and a consistent user experience, FirebaseUI is the recommended approach.It provides a drop-in authentication solution with pre-built UI flows for Google Sign-In, handling the entire user interface for login and registration. Alternatively, a custom UI can be built using the Firebase Authentication SDK for more granular control over the look and feel.
- **API Schema Design:** The primary interaction with the backend will be through the Firebase Authentication REST API. When a user authenticates, the client application receives a JSON Web Token (JWT).This token is then sent in the authorization header of subsequent API requests to protected backend resources. The backend, in this case, can be Cloud Functions or another serverless solution that verifies the JWT to authorize the request. The user information can be accessed from the decoded JWT on the backend.

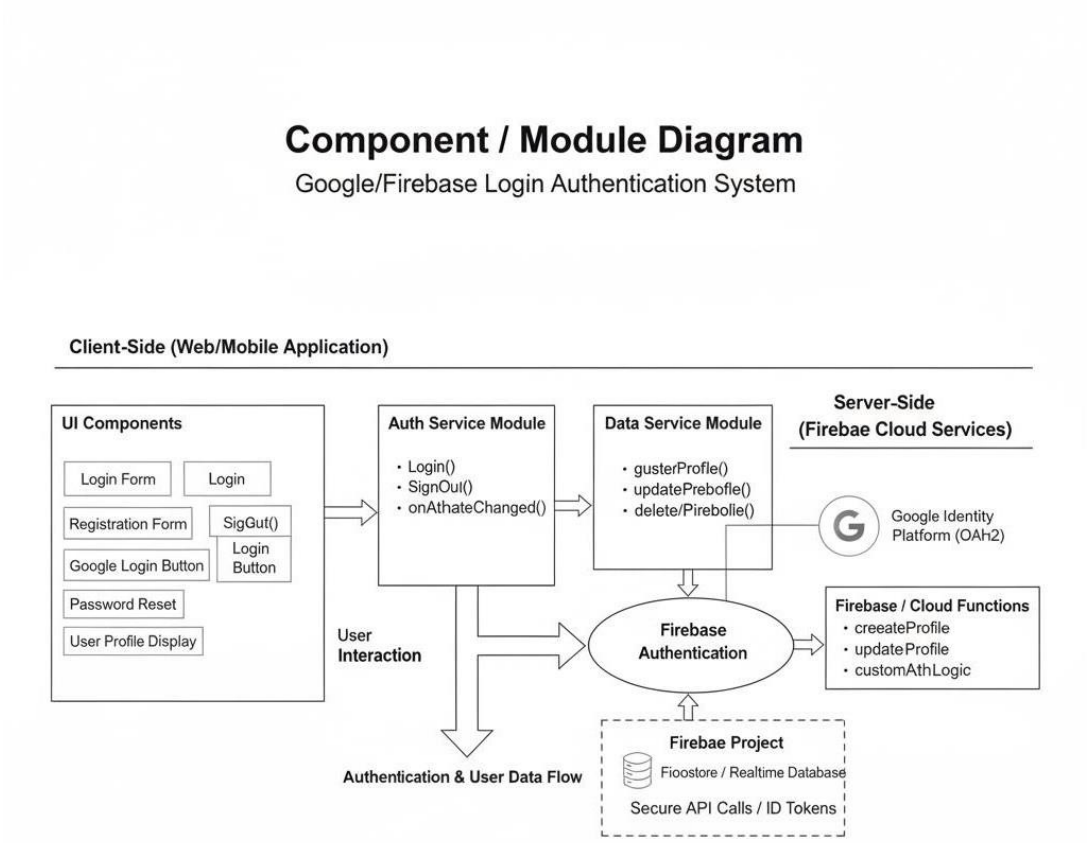
.

Data Handling Approach

The data handling approach prioritizes security and efficiency, ensuring user data is managed responsibly.

- **User Authentication Data:** Firebase Authentication securely stores and manages user credentials, including information retrieved from Google upon successful login (e.g., name, email, profile picture).
- **Application-Specific User Data:** Additional user information will be stored in Cloud Firestore or Realtime Database. The user's unique Firebase UID, obtained after authentication, will be used as the primary key to associate this data with the authenticated user.
- **Data Flow:** Upon successful Google Sign-In, Firebase provides the client with the user's profile information and a JWT. This JWT is then used to authenticate requests to the application's backend. The backend can then use the user's UID from the verified JWT to query the database for any additional required data. This approach avoids sending sensitive user data with every request and relies on a secure token-based authentication mechanism.

Component / Module Diagram



Basic Flow Diagram