**COLLEGE CODE : 9623**

**COLLEGE NAME : Amrita College of Engineering And Technology**

**DEPARTMENT : Computer Science and Engineering**

**STUDENT NM-ID : BD196A9C241DE258504454E368A99484**

**ROLL NO : 23CS058**

**DATE : 25-09-2025**

**Completed the project named as**

**Phase 3 MVP Implementation**

**PROJECT NAME : LOGIN AUTHENTICATION SYSTEM**

**SUBMITTED BY,**

**NAME : MITHRAN G J**
**MOBILE NO : 8825427625**

## Phase-3 MVP Implementation:

### 1. Project Setup

Steps to create and configure the Firebase project:

1. Create a new Firebase project in the Firebase Console.

2. Enable Authentication providers such as Email/Password and Google sign-in.

3. Create a Firestore database (start in test mode for MVP, then secure rules before production).

4. Add a web app to the Firebase project to get the firebaseConfig object and SDK snippets.

5. Include Firebase in your HTML using script tags provided by Firebase.

### 2. Core Features Implementation

The MVP includes the following pages:

- Login Page (index.html)
- Signup Page (signup.html)
- Dashboard Page (dashboard.html, protected area after login)

Core authentication features include:

- Email/Password Sign-Up and Sign-In
- Google Sign-In
- Sign-Out option
- Route protection using authentication state listeners

## 3. Data Storage (Local State / Database)

Local State: Temporary UI-related state can be stored in memory or localStorage/sessionStorage for minor preferences. Sensitive data should not be stored in local storage.

Database: Firestore is used to store user profiles and application data. For MVP, a simple structure with a "users" collection is sufficient. Each user document contains details like display name, email, and timestamps.

Security: In MVP, rules may start in test mode, but for production, Firestore security rules should restrict access to authenticated users only.

## 4. Testing Core Features

Manual testing is used to validate features in MVP. Key test cases include:

- Successful sign-up with valid email and password

- Successful sign-in with existing credentials

- Error messages for incorrect password or invalid credentials

- Google Sign-In functionality

- Sign-out functionality leading to restricted access to protected pages

Edge cases include handling duplicate sign-ups, network failures, and incorrect inputs.

Automated testing can be introduced later, but manual testing is sufficient for MVP.

## 5. Version Control (GitHub)

A GitHub repository is used to manage the project. Suggested structure:

- Public folder containing HTML, CSS, and JS files

- README file with setup instructions

- .gitignore file to exclude unnecessary or sensitive files

Best practices include:

- Using feature branches for development

- Writing clear and concise commit messages

- Creating pull requests for review before merging changes

- Tagging releases such as v0.1.0 for the MVP version