# System Design - Version 1

## Architecture Overview

Version 1 introduces a backend system with an admin login and cloud storage. The system consists of three main components:

1. ️Frontend (User Interface) – Displays news posts to users.
2. Backend (Authentication & Database) – Handles login, stores posts, and manages data.
3. Deployment (Hosting & Database Service) – Hosts the app and stores data online.

## System Components & Workflow

#### Admin Login System (Authentication)

1. Purpose: Only admins can access the dashboard to add/edit posts.
2. Technology Used: Firebase Authentication (Google Sign-In for simplicity).
3. Process:
   1. Admin enters credentials (Google Sign-In).
   2. Firebase verifies authentication.
   3. If successful, the admin is redirected to the dashboard.
   4. Unauthorized users are blocked.

#### Online Database (Post Storage & Retrieval)

1. Purpose: Store news posts permanently in a cloud database.
2. Technology Used: Firebase Firestore (NoSQL Database).
3. Process:
   1. Admin adds a post → Data is saved in Firestore.
   2. Users load the app → Posts are retrieved from Firestore.
4. Data Structure (Firestore Collection):

{

"posts": [

{

"id": "12345",

"title": "Top 10 Productivity Apps",

"bulletPoints": [

"App 1 - Description",

"App 2 - Description"

],

"summary": "A quick overview of the best productivity apps.",

"source": "https://example.com",

"tags": ["productivity", "apps"],

"date": "2025-02-06",

"author": "Admin Name"

}

]

}

Picture view:



#### Admin Dashboard

1. Purpose: Allow admins to manage posts via a private panel.
2. Technology Used: HTML, CSS, JavaScript (Same as before).
3. Features:
   1. Add, edit, delete news posts.
   2. View all stored posts.

#### User Interface (Frontend)

1. Purpose: Display news posts dynamically from Firestore.
2. Process:
   1. The homepage fetches posts from the database.
   2. Users can filter posts by date or tags.

## Deployment & Hosting Plan

|  |  |  |
| --- | --- | --- |
| **Component** | **Service Used** | **Why?** |
| **Frontend** | GitHub Pages | Free & easy to deploy |
| **Authentication** | Firebase Auth | Free Google Sign-In |
| **Database** | Firebase Firestore | Free tier available |

## System Flow Diagram (Logical Flow)

1. Admin logs in
2. Admin adds news post
3. Data is stored in Firestore
4. Users see news on the website