Team Members:

Tanishq Agarwal(Team Lead)
Rupam Chakraborty
Abhinash
Darshini P N
Kasturi Satwik
Pradeep Kumar

Submitted To: Mr. Yogesh Raje

Team ID: AICB1-CC02

Team Name: Pegasus

Week 1 Problem Statement:

Background of the problem:

This project's implementation target solution is to transfer the deferred manual attendance systems to newly adopted cloud technology based attendance maintenance systems. It is designed to reduce error rates and to make error correction much more easier than manual attendance based systems. Cloud attendance systems ensure data duplicacy is avoided wherever possible.
The most important reason to shift this particular solution is to make the data redundant in nature and to make it's accessibility device independent. Also scalability is a feature which manual attendance systems fail to provide which is fixed by introducing this feature in cloud based attendance managers.

Customer:

1. Corporations and Multinational Companies who have an abundance of employees and their attendance data to be maintained on a daily basis.
2. Government require such platform applications in order to digitize their employee's attendance data.
3. Educational institutions can also use this platform to maintain the attendance of their students and faculty staff.
4. Indian Railways Reservation Ticket Collectors.

Problem Statement:

To develop a Cloud based solution for an Attendance Management platform using PaaS (Platform as a Service).

Week 2 Solution Development:

Solution Background:

We would like to develop a cloud based solution in the form of a full stack website which can store attendance data in the form of a database in the cloud. The data from the database is retrievable from the cloud using API gateways working in the background.

Proposed Solution Implementation Details:

1. We will firstly try to implement our website design in a UI work-flow space like **Figma** or Adobe XD (we will make a wire frame of our solution in Figma for this project.)
2. The second thing to do is to implement the website on a platform in the form of a framework for example React or Angular. We will be using **React Native** for this project as our development framework.
3. The next thing we need to perform is to transform our designed work-flow into a Web-Package and to create a Database, in which we will create a User Pool to store our users(whether registered ones or non-registered ones) and their details. For this purpose we will be using something called as **GCP Firebase Firestore** service which helps us provide authorization services using embedded API's of GCP Firebase.
4. **We will be using the following PaaS services available in :**
   - **Google Cloud Platform GCP Firebase Hosting**
   - **Google Cloud Platform GCP Firebase Firestore**
5. The solution will be divided into Testing and Production stages. The development of the solution will be focused in the Testing stage in the local environment whereas the final Solution i.e. the Production Stage will be implemented.
6. The solution designing starts when we design the page elements in React Framework and then publish it on Firebase Hosting which is a deployment service provided by Google Cloud Platform.
7. Google Cloud Platform Firebase Hosting is a fast and reliable deployment web-tool based on cloud which focuses on deploying user websites on the web whose up-time is very good and the management of which is extremely efficient. Google Firebase Hosting supports various ways of after deployment updation techniques for example Deployment update using Local files and from Firebase CLI available to download for free and use on any system. The hosting can be pushed and committed to GitHub repositories.
8. Our project's entire React framework page will be deployed using GCP Firebase Hosting on cloud. The website will have several features along with a Login and Sign

Up page fully backed up by services from **Google Cloud Platform's Firebase Authentication API's.**

9. The management will contain **features** like:
   Attendance , User List Portal and Session Logout and Login management using API's from Google Firebase Auth.
10. The final product will be a full fledged website with Attendance Management services deployed on a cloud provider (**in our case Google Cloud Platform Firebase**), whose description will be shared on the Git Hub page hosted by the team at the end of Week 4.

Uniqueness in the Solution as compared to existing solutions :

1. **Manual Attendance(Pen and Paper system):**
   The existing solution available and most widely used in India and other nations to manage attendance is to use pen and paper. This project seeks to change this approach by implementing a fully soft version of this solution where data manipulation and editing becomes much more manageable also while being able to save paper and being eco friendly.

2. **Soft Attendance (using Excel and Access):**
   This type of attendance management has been recently introduced in the newer educational institutions in India and it uses computers to store attendance data locally on the systems available on the premises of the institution. It is not very efficiently managed or even backed up, so data loss is a continuous threat to such data.

3. **Cloud Attendance(using built in GCP Firebase and Firestore API implementation):**
   Existing cloud based solutions that have been implemented for this problem have been using PaaS services by simply creating a PHP and HTML–CSS based web page which fetches data from a plain PHP server hosted on the cloud. In this project we will be designing the framework such that our Data is fetched very securely from the DBMS server using an API developed by Google for Firebase Firestore Database users. This is a unique solution to the implementation of the project. The API will use methods to access data from the server such as username and session management services along with user verification Auth codes. This will maintain complete security over the project and its users who have access to the data from the website.

Week 3: Solution Design

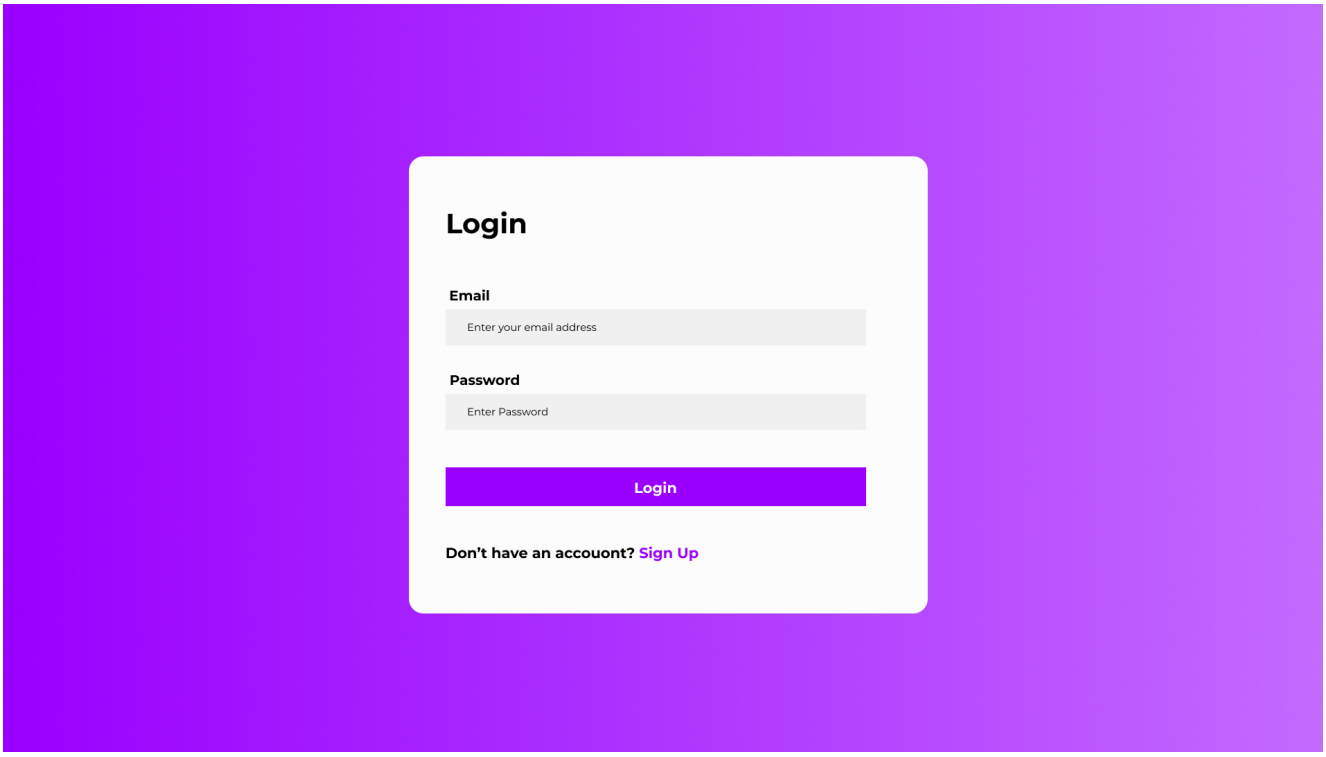
Solution Design Background:

The solution design implemented by our team includes **wireframes, userflows and sitemaps** for the user to recognize and implement the solution design. The solution design by our team

Wireframe:

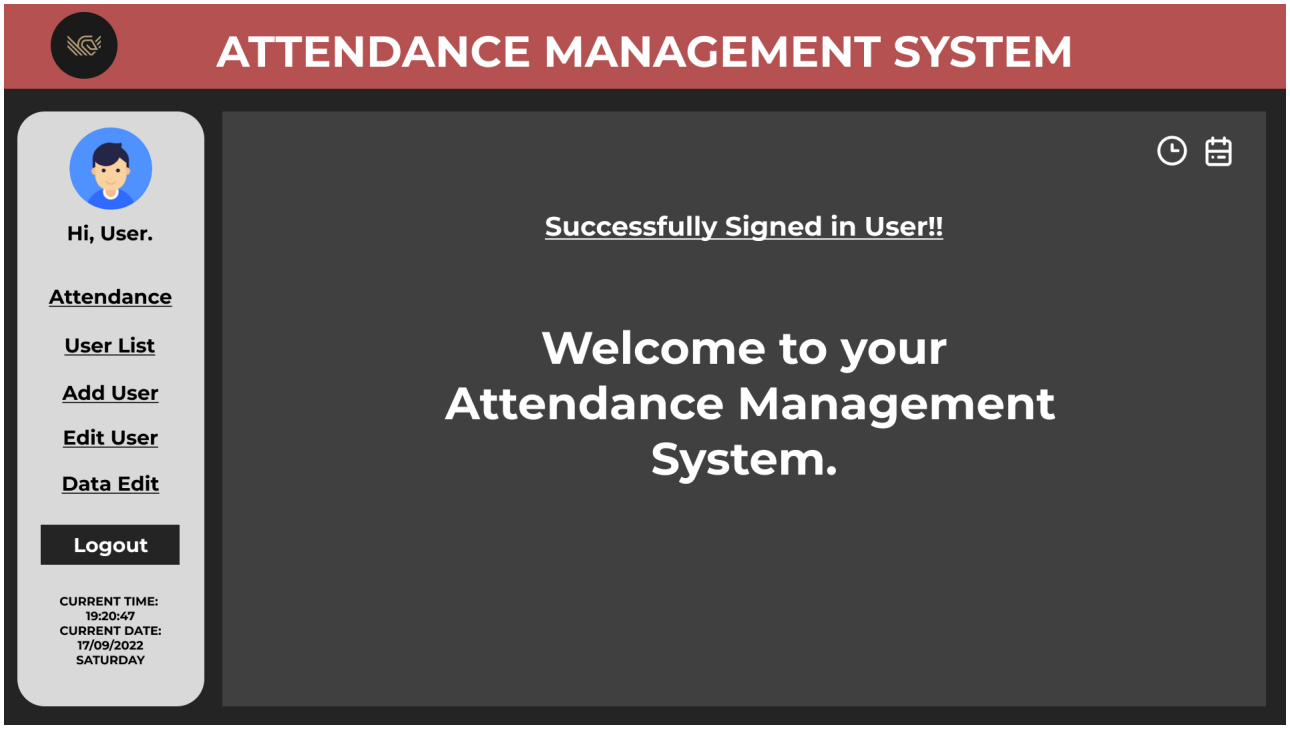
The wireframe will consist of user login page and the main home page along with feature pages like Attendance Page and User list page. The data flow is maintained in each page and it is represented below in form of screenshots of the wireframe designed by the team:

1. **Login Page:**



- The page is designed in a very minimalist manner to keep it simple, yet very appealing. The sign-up and sign-in features are fully backed by services from Firebase and its hosting service.
- This page is deigned using CSS and React-Native.
- The page is the access door to the entire web application and is the first view of the user while using the application.

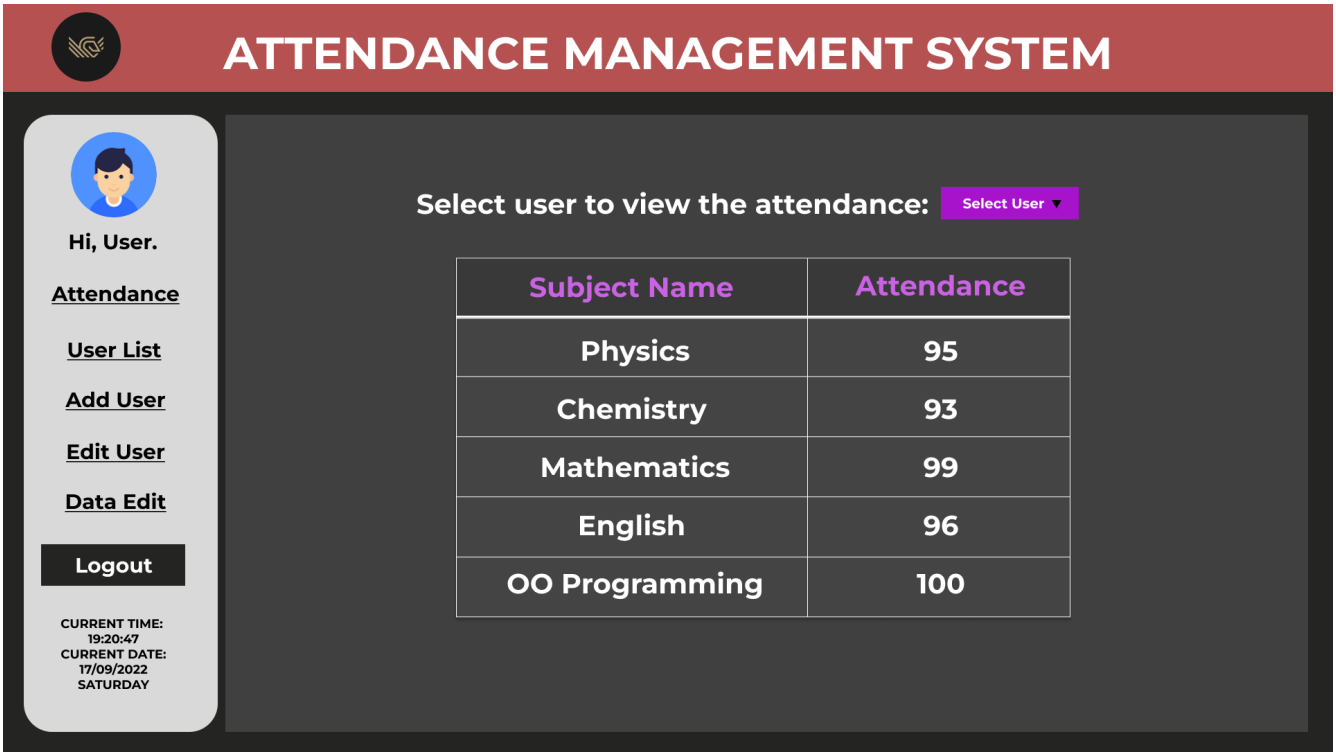
2. **Attendance System Home Page:**



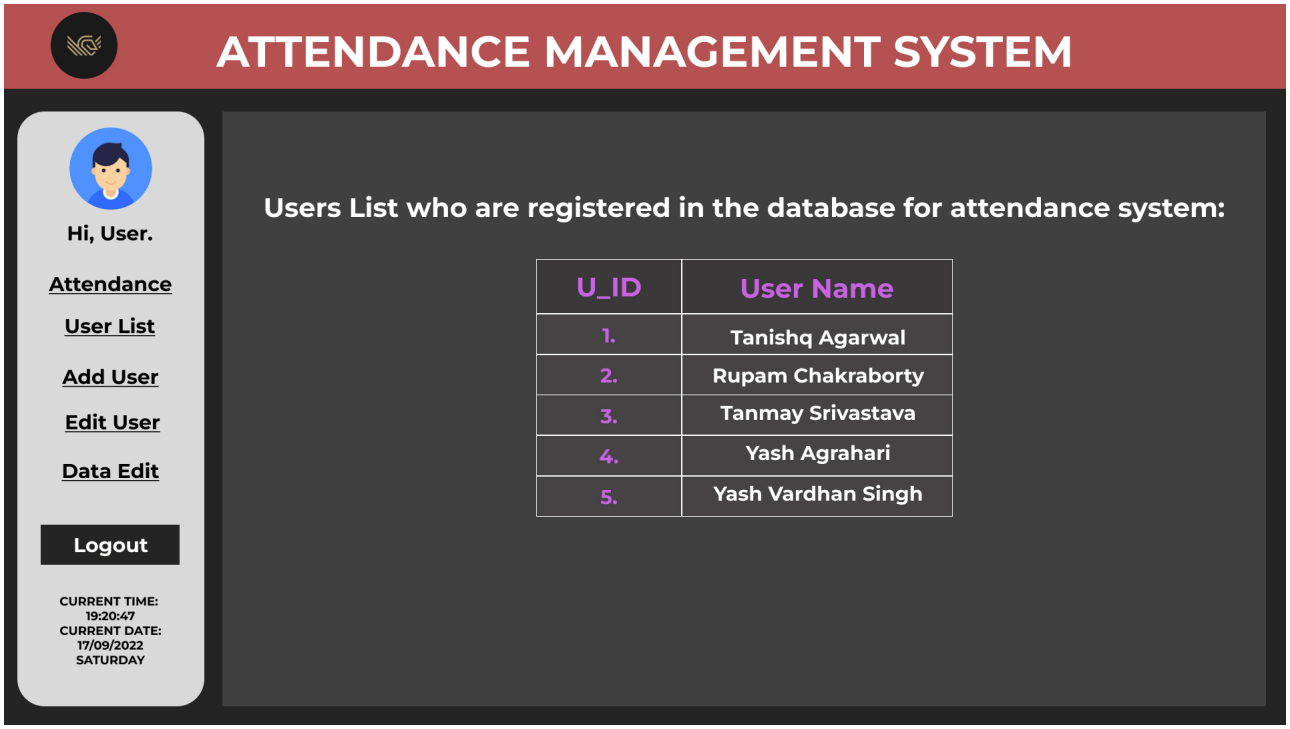
- The data on this particular home screen page is obtained from the Firebase Firestore when the user is authenticated and the obtained data is reflected as username on the left panel of the web-page.
- The design of the front page is kept very simple while keeping the user easiness in mind. The data for registered users can be accessed on this portal's right hand side main app window which acts as a panel for other features available on user panel to be projected while keeping the same interface so that website does not have to change pages while loading again and again. This is achieved using React Router DOM.

3. **Attendance Management Page:**

- The page is designed to provide a tabular view of the attendance data of the authenticated user.
- Data is fetched from Firebase Firestore Database (already deployed by the Admin) using inbuilt API's provided by Firebase itself for their CRUD (Create, Read, Update and Delete) applications.
- The data is then modified to be presented in form of a table that individually represents the data of the user for each domain in which his attendance is required.
- The page also supports attendance view for all the users available on the Firebase Firestore DB.

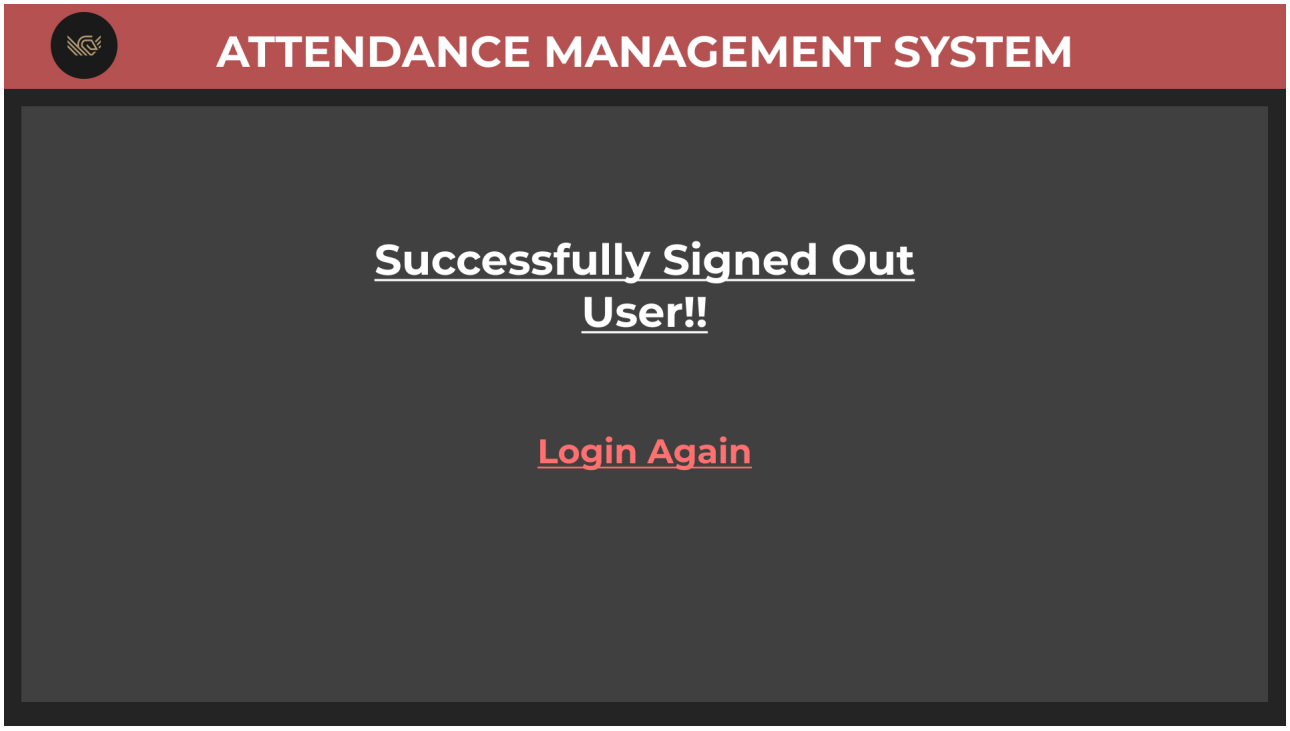


4. **User List Page:**



- The data is available for many users on the database, the names of whom can be displayed on this page as an app feature. This page mainly focuses on providing the names and ID's of the registered users of this app and whose data is available to be fetched.

5. **Logout Page:**

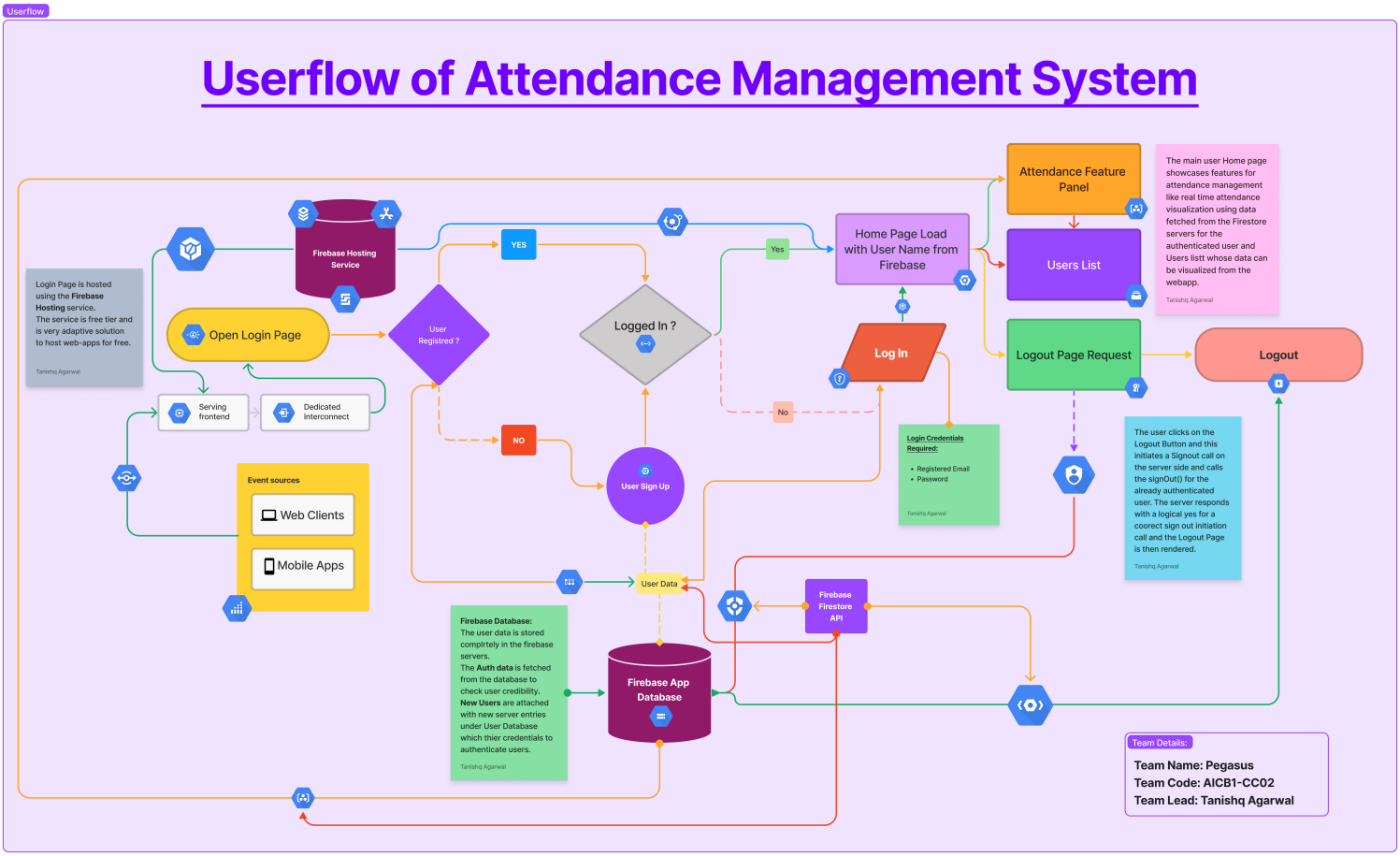


- This page is presented to users who have successfully logged out of the web-app using the Logout button available to them on the left panel of the web-app.
- The Logout button initiates a chain of Logout sequences starting with the calling of the signOut() method from the API and ending with a successful logout message to the user in form of this page on the screen at last.

That's all regarding the Wireframe of the web-app to be designed. The upcoming pages in this document present the Userflow using flowcharts designed to depict the user data flow which is provided by the user and which is fetched from the Firebase Firestore DB's.

The data flow diagram given below as a userflow dagram is made with full support of icons from Google Official Firebase icon set and each icon represents the cloud services used for that particular operation/feature in the web-application.

Userflow:



- The above user flowchart explains how the data flows between various segments of the application's front-end and the back-end.
- The data flow is successfully depicted using the flowchart arrows and icons in their respective positions. The data is properly flown inside the application using React and JS on the backend logic. The data integrity is maintained in the entire application using API's from Firebase. The database opens up an active monitoring application programming interface (API) session each time a data fetch request is made or a data entry or submit is made from the application.
- A note is also added to each entry which is complex to explain using the userflow chart and demands greater level of explanation.