# Patient Visit Manager - Backend & Frontend Report

## 1. Backend Status

The backend of the Patient Visit Manager project has been fully developed and tested. It is exposed through a .NET Core Web API and provides all the required functionalities including:

- Patient CRUD operations

- Doctor CRUD operations

- Visit scheduling and querying

- Fee schedule retrieval

- Activity log retrieval

- User authentication and authorization (JWT-based)

The backend is fully functional and can be accessed via Swagger for testing and interaction. Swagger is hosted at:

http://localhost:49428/swagger/index.html

## 2. Authorization & Authentication

Authorization and Authentication mechanisms have been successfully implemented in the backend. Users are required to login with valid credentials to access protected endpoints. JWT (JSON Web Tokens) are issued upon successful login, containing role-based claims. Roles include Admin, Doctor, and Receptionist, with access restrictions enforced accordingly.

## 3. Frontend Status

A simple HTML, CSS, and JavaScript frontend has been prepared under the 'wwwroot' directory. The frontend connects to the backend API for performing CRUD operations and displaying data.

## 4. Current Issues in Frontend

While the backend is working seamlessly, the frontend has some limitations and unresolved issues:

- Due to time constraints, the Sign-Up (User Registration) page could not be developed.

- Add functionality for Patients, Doctors, and Visits could not be fully achieved or tested properly in front end but in backend all the controllers work fine and have been tested properly.

- Some requests from the frontend may not integrate correctly with the backend without adjustments.

## 5. Conclusion

The backend API is fully functional, robust, and well-documented through Swagger. Authentication and Authorization systems are working as intended. The frontend demonstrates connectivity with the backend but requires further refinement, especially in implementing the user registration (Sign-Up) functionality and ensuring proper handling of Add operations.

# IMPORTANT NOTE: -

**To sign up a user, it has to be done via backend controllers as the sign up page is not there in front end.**

**For simple testing you can login as**

**Username:** [**admin@example.com**](mailto:admin@example.com)

**Password: admin123**

**As these details are stored in the DB with Hash and Salt passwords**

**After Updating the data in the front end, reload the website to see the updated changes. If any error pops up please ignore it.**