**BloodLink — Project Reflection Report**

**1. Inspiration & Problem Statement**

Every time I had to visit the hospital for routine blood checks, I needed to arrive as early as 7:30 AM just to get in line. Despite arriving early, I often wouldn’t finish until 2 or 3 PM. Waiting in long queues, often while feeling unwell, made me realize how inefficient and exhausting the current manual system is. This personal experience motivated me to develop BloodLink — a system that could reduce waiting times and make the process smoother for patients like me.

**2. Key Features**

BloodLink includes several important features designed to solve these problems:

* **Digital Token System:** Helps manage patient flow and eliminates the need to physically stand in long queues for hours.
* **Real-Time Lab Report Sharing:** As soon as lab results are updated, they become instantly accessible to doctors and nurses, reducing delays in diagnosis and treatment.
* **Digital Patient Records:** Patient data is securely stored and easily managed, allowing faster access and updates without relying on physical record books.

**3. Technology Stack**

The project uses:

* **React** for building a dynamic and responsive frontend user interface, styled with CSS.
* **Java Spring Boot** for backend APIs and business logic.
* **MySQL** as the relational database to store patient records, lab reports, tokens, and other system data.  
  This stack was chosen because it allows for scalable, maintainable development with a strong separation of frontend and backend concerns.

**4. User Roles**

BloodLink is built with a well-structured role-based access system ensuring that each user accesses only the data relevant to their role:

* **Doctor:** Views patient details, updates diagnoses, prescribes medicines, and accesses lab reports.
* **Nurse:** Handles patient registration and manages the digital token system to streamline queues.
* **Lab Technician:** Updates lab test results and uploads lab reports that instantly become accessible to doctors.  
  This approach enhances security and efficiency by sharing only the necessary information with each user.

**5. Challenges**

One of the biggest challenges was managing data flow between React components — especially passing data between deeply nested or dynamic components. I had to learn effective use of props, lifting state up, and eventually React Context to manage global state. This improved the frontend’s maintainability and user experience.

**6. Real-World Use**

I believe BloodLink has the potential to be used in real government hospitals. However, much more research and development are needed. I built the system based mainly on personal experience and observation. To make it production-ready, I would need to:

* Study hospital workflows in more detail
* Implement stronger data security measures
* Ensure scalability for large user volumes
* Integrate with existing hospital systems and comply with health authority regulations  
  With these steps and professional feedback, BloodLink could evolve into a reliable, production-level tool.

**7. Lessons Learned**

I learned that even in countries like Sri Lanka, where many systems remain basic and outdated, technology can still provide meaningful improvements. While global tech advances rapidly, local challenges persist. This project taught me that creating relevant, practical solutions tailored to local needs is just as important as using the latest technology.

**8. Future Plans**

I plan to integrate an **AI agent** into BloodLink to enhance its functionality. The AI could automate routine tasks such as patient triage, predict blood supply needs, assist doctors with decision support, and provide patients with real-time answers to common questions. This would transform BloodLink from a digitization tool into a smart, proactive system that further improves efficiency and patient care.