# Lab 1 Report: IDE Setup and Validation

Course: IoMT-Based Stress Monitoring System  
Student: Miltone  
Institution: Michigan Technological University  
Instructor: Dr. Ronghua Xu  
Duration: Week 1  
Lab Title: Setting Up the Development Environment for IoMT Application Frontend  
Objective: To establish, configure, and validate the integrated development environment (IDE) required to build the Flutter-based frontend of the IoMT stress monitoring system.

## 1. Introduction

The first step toward any robust software system is constructing the environment where imagination meets execution. In this lab, the goal was to set up a reliable Flutter development workspace—the foundational layer on which every subsequent module (UI design, backend integration, API connectivity) of the IoMT stress monitoring system would be built.  
  
This phase represents the genesis of the entire project lifecycle: configuring tools, validating dependencies, and ensuring smooth cross-platform compatibility between the developer’s system, Android emulator, and cloud environments that would follow.

## 2. Objectives

By the end of this lab, the following deliverables were to be achieved:

* Install and configure Flutter SDK and Dart SDK.
* Set up VS Code and Android Studio for Flutter development.
* Validate environment setup using the flutter doctor diagnostic tool.
* Create and execute a default Flutter app across both web and Android emulator environments.
* Capture and document the process with screenshots and a short demo video.

## 3. System Setup and Configuration

Hardware and OS Environment:

|  |  |
| --- | --- |
| Operating System | Windows 11 (64-bit) |
| Processor | Intel Core i7 |
| RAM | 16 GB |
| Emulator | Pixel 6 Pro (Android 13) |
| Browser | Google Chrome (Flutter Web) |

These specifications ensured optimal performance for Flutter compilation and virtual device execution, preventing build delays or emulator lag.

## 4. Installation Steps

Step 1: Installing Visual Studio Code

Installed VS Code from official website and added Flutter and Dart extensions. Configured the SDK path and verified Flutter integration through Command Palette.

Step 2: Installing Android Studio

Installed Android Studio, integrated Flutter and Dart plugins, and configured Android SDK path. Created Pixel 6 emulator through AVD Manager.

Step 3: Installing Flutter SDK

Downloaded Flutter SDK, unzipped to C:\src\flutter, and added to PATH environment variables. Verified installation using 'flutter --version'.

Step 4: Running Diagnostic Tool

Executed 'flutter doctor' to confirm installation. Output showed all components configured successfully.

Step 5: Validating the Setup

Created a test Flutter project ('flutter create test\_app') and ran it on both Android emulator and Chrome browser. The default counter app successfully displayed incrementing counter text, confirming setup integrity.

## 5. Architecture of the Development Environment

Flutter SDK (Local)  
│  
├── VS Code (Development IDE)  
│ └── Dart Compiler  
│  
├── Android Studio (Emulator + SDK)  
│  
└── Chrome Web (Cross-platform test interface)

## 6. Challenges and Resolutions

|  |  |  |
| --- | --- | --- |
| Challenge | Description | Resolution |
| PATH variable not detected | Flutter commands unrecognized initially | Re-added system PATH and restarted terminal |
| Android SDK missing | Flutter doctor reported 'No Android SDK found' | Installed SDK via Android Studio setup wizard |
| Emulator lagging | Emulator performance issues | Enabled hardware acceleration (Intel HAXM) |
| VS Code extensions inactive | Flutter extension failed to auto-load | Reinstalled Dart and Flutter extensions manually |

## 7. Outcome and Deliverables

Final deliverables achieved:

* Flutter SDK and Dart installed and verified.
* VS Code and Android Studio configured successfully.
* Flutter Doctor report passed with zero issues.
* Default counter app executed successfully on emulator and web.
* Screenshots and demo video submitted.

## 8. Reflection

The success of this lab wasn’t just technical — it was philosophical. In this configuration process, patience met precision; every installation step symbolized the discipline needed in scientific computing. The system was now alive — a clean environment, humming quietly, ready to transform data into insight, and insight into care.

“Before the code could breathe, the ground had to be tilled. In this first lab, I didn’t just install software — I built the soil for digital healing to grow.”

## 9. Forward Outlook

This environment now supports:  
- Cross-platform app testing (Android & Web)  
- API integrations (Fitbit, AWS)  
- Real-time cloud synchronization (to be achieved in later labs)  
  
With this robust foundation, the path ahead—spanning UI design, AWS Lambda, and EventBridge automation—stands ready to converge into the full IoMT stress-monitoring ecosystem.

✅ Lab 1 Successfully Completed. The IDE is configured, the SDKs are alive, and the environment awaits the next heartbeat of development.