**Project Handover: Smart Glasses - Phase 1**

### Date: 26th January, 2025

# Project Overview

The Smart Glasses project integrates wearable technology with mobile connectivity to create an unobtrusive information display system. Using an ESP32 microcontroller and OLED display mounted on standard eyewear frames, the device shows real-time information in the user's peripheral vision.

# Core Functionality

The system consists of three main components:

**Hardware Layer:** ESP32-based smart glasses with OLED display

**Mobile Interface:** React Native app handling notification relay and status relay

**Communication Layer:** MQTT-based messaging system for real-time notification updates

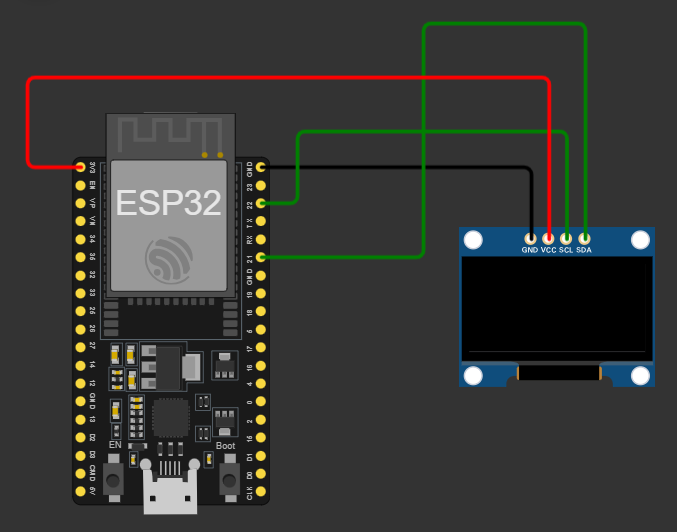
# Key Features & Implementation

**Weather Snapshots**: Temperature and weather information along with location and date are displayed from weather API.

**Notification Forwarding**: Phone notifications from all applications including call alerts forwarded to OLED display.

# Technical Architecture

The following image below describes the ESP32 and OLED Display connections:



# Tools and Technologies:

* **Android Studio**  
  Android Studio is an official Integrated Development Environment (IDE) for Android app development. It provides tools to build, test, and debug Android applications. In this project, Android Studio might be used alongside React Native for native feature integration, debugging the app, and packaging it for deployment on Android devices.
* **React Native**  
  React Native is a JavaScript framework used for building cross-platform mobile applications, including the Android app in this project. It allows developers to write code once and deploy it on multiple platforms, ensuring a consistent user interface and functionality. In the Smart Glasses project, React Native can be used to create the companion Android app for interfacing with the IoT device, providing real-time updates, control features, and data visualization.
* **MQTT Protocol**  
  MQTT (Message Queuing Telemetry Transport) is a lightweight messaging protocol optimized for low-bandwidth and high-latency networks, ideal for IoT devices like ESP32. In this project, the ESP32 uses MQTT to send sensor data (e.g., from the glasses) to the broker and receive commands from the Android app. The broker acts as a central hub to facilitate communication between the IoT device and the app, ensuring real-time data transfer and control.
* **MQTT Messaging Brokers**

MQTT brokers are central components in MQTT-based systems, acting as intermediaries that manage and route messages between publishers (e.g., IoT devices like ESP32) and subscribers (e.g., Android apps).

* **Arduino IDE**  
  Arduino IDE is a development environment used for programming microcontrollers like the ESP32. In the Smart Glasses project, the Arduino IDE is used to write and upload the firmware to the ESP32. This firmware handles sensor input, MQTT communication, and control signals to and from the Android app, ensuring seamless operation of the IoT device.