PROJECT ABSTRACT

Wi-Fi Network Scanner Desktop App

A desktop Wi-Fi scanner with signal analysis in dBm, channel utilization graphs, security auditing and CSV export. Delivered built-to-order with the full application, installer, site-survey guide, report, PPT and viva kit.

01. Title

Wi-Fi Network Scanner Desktop App — discover nearby access points and analyze them like an engineer: signal in dBm, channel and band, security suite, vendor OUI and per-channel congestion graphs.

02. Introduction / Background

Slow Wi-Fi in crowded buildings is usually a congestion problem: dozens of access points piled onto the same channels, invisible to users who only see signal bars. A proper scanner exposes the radio facts — SSID, BSSID, RSSI in dBm, channel, band, width, security suite, vendor — and aggregates them into channel utilization views that make the least-congested channel obvious. It also audits security at a glance, flagging open networks and legacy WEP.

03. Problem Statement

Consumers cannot diagnose Wi-Fi problems with bars alone, and professionals need survey data for channel planning. This project builds a strictly passive desktop scanner: no deauthentication, no cracking — only reading the beacon frames access points already broadcast. It presents signal history, neighbor analysis and CSV export for site surveys.

04. Objectives

- Discover access points across 2.4 GHz and 5 GHz with SSID, BSSID, RSSI, channel, band and security parsing.
- Grade signal quality from dBm and resolve vendors from MAC OUI prefixes.
- Render per-band channel utilization graphs for congestion analysis.
- Track per-network 60-second RSSI history with co-channel neighbor lists.
- Export scans to CSV; document a worked channel-planning site survey.

05. Proposed Methodology

The app drives the wireless adapter through channel-by-channel sweeps, collecting 802.11 beacon and probe-response frames. Scapy parses each frame for SSID, BSSID, RSSI, channel, width and the RSN information element describing security. A local OUI table resolves vendors. Results aggregate per channel into utilization graphs weighted by signal strength. Selecting a network starts an RSSI sampler for the history plot and neighbor analysis.

06. System Architecture / Working

Adapter scan control → frame capture → 802.11 parser (Scapy) → network table model → OUI resolution → channel aggregator → Qt UI (network list, detail with RSSI history, channel graphs) → CSV exporter + SQLite scan history.

07. Hardware / Software Requirements

- A Linux machine (richest support) or Windows/macOS with a scan-capable Wi-Fi adapter.
- Python 3, PyQt6, Scapy, SQLite, OUI database, PyInstaller.
- Monitor/scan-capable adapter; 6 GHz needs a 6E-capable adapter.

08. Expected Outcomes

A working passive Wi-Fi scanner with dBm analysis, channel graphs, security flags and CSV export — plus a site-survey walkthrough, project report, PPT and viva Q&A.

09. Applications

- Home users picking the least-congested channel for their router.
- Students learning 802.11 fundamentals with real frames.
- Small offices doing before/after surveys when repositioning access points.

10. Future Scope

- Long-term RSSI logging for coverage heatmaps.
- Rogue-AP anomaly alerting on duplicate SSIDs.
- Integration with floor-plan overlays for walk-through surveys.