# Laptop Support System Project Documentation

Muhammad Al-Zubair Obaid, Muhammad Haj Eesa, Muhammad Al-Sattouf, Muhammad Mahmoud, Mosa Othman, Abd-Allah Ghzail

## Overview

This project aims to enhance the functionality and performance of personal computers by providing features such as displaying PC information, optimizing the PC, automating and manually installing drivers, registering laptop data, implementing object-oriented programming (OOP) classes, and storing data in SQLite.

## Features

### Show PC Information

This module gathers and displays comprehensive information about the PC, including:

- **System Specifications:** Processor type, RAM size, GPU details, and storage capacity.
- **Operating System Details:** Version, build number, and architecture.

### Optimizing PC

This feature focuses on improving the PC's performance through:

- **Disk Cleanup:** Removing unnecessary files and temporary data.
- **Memory Management:** Optimizing RAM usage by closing background applications.
- **Startup Optimization:** Managing startup programs to decrease boot time.
- **Defragmentation:** Organizing the hard drive to improve read/write speed.

### Automatic and Manual Driver Installation

This module allows users to install drivers effortlessly:

- **Automatic Installation:** Detects outdated or missing drivers and installs the latest versions automatically.
- **Manual Installation:** Provides an interface to manually select and update drivers from a list.

### Laptop Data Registration

This feature enables the registration and management of laptop data:

- **User Information:** Capturing user details such as name, contact information, and device ID.
- **Laptop specifications:** Hardware properties of the laptop.

## Object-Oriented Programming (OOP) Project Classes

The project is structured using OOP principles, ensuring modularity and scalability:

- **Class Structure:**
    - **Driver:** (Properties: Component, Version, Release Date, Size)
    - **Laptop:** (Properties: Manufacturer, Model Family, Model, CPU, Integrated GPU, Dedicated GPU, RAM, Storage, Storage Type)
    - **UserDetails:** (Properties: Email, Username)

**Key Components**

**1. Server Setup**

- Creates a local HTTP server on port 8050 using `HTTPServer`

- Handles API requests through the `RequestHandler` class

- Runs in a separate thread to avoid blocking the main UI

**2. MainWindow Class**

The main application window class that inherits from QMainWindow. Key functionality:

- Initializes the UI from the Qt designer file

- Starts the local server

- Connects UI signals to handlers

- Handles web view loading and progress

**3. System Information Collection**

When the web page loads, it:

- Uses WMI (Windows Management Instrumentation) to collect detailed system information:

- Computer manufacturer and model
- CPU details (name, speed, cores)
- GPU information (integrated and dedicated)
- RAM configuration
- Storage devices
- Windows version details

**4. Data Population**

- For index.htm: Populates PC specifications into the webpage
- For contribute.html: Autofills the contribution form with system details
- For drivers.html: Previews available drivers to install automatically or manually

**5. Data Flow**

1. Application starts and creates local server

2. Main window loads with web view

3. When web pages load, system info is collected via WMI

4. Data is formatted and injected into web pages using JavaScript and HttpRequest

5. UI updates reflect the current state and loading progress

The application serves as a bridge between the system's hardware information and the web-based user interface, providing a seamless way to display and collect system specifications.

## Conclusion

This project leverages advanced technology to improve PC performance and management. By utilizing OOP principles and SQLite for data handling, it offers a robust solution for both novice and experienced users. Continuous updates and enhancements ensure that the software remains relevant and efficient in meeting user needs.