



Queensland
Government

Queensland Health

Statewide Food Safety Auditor Registration Project

Developer Handover Documentation

QLD Health Food Safety Auditor Search - Developer Handover Documentation

Project Overview

The QLD Health Food Safety Auditor Search is a simple client-side web application that provides a searchable interface for Queensland's approved food safety auditors. The application parses Excel data from a hosted source and presents it through an interactive filtering system.

Technology Stack

- **Frontend:** HTML5, CSS3, JavaScript (ES6+)
- **Styling:** Queensland Government Design System (QGDS), Bootstrap 5
- **Data Processing:** SheetJS (xlsx library)
- **Deployment:** Static hosting (any web server)

Project Structure

```
QLD-Health-Capstone-T302/  
├─ index.html           # Main application page  
├─ script.js            # Core application logic  
├─ assets/              # Images and other static assets  
└─ css/  
    ├─ styles.css       # Custom styling  
    └─ qg-main.css      # QGDS main styles
```

Installation and Setup

Quick Start

- Install Node.js and npm
- Install live-server globally
- Clone and navigate to project directory
- Start development server
- Open browser to verify setup

Prerequisites

Required Software

- Node.js (v16.0.0 or higher)
 - o Download from: <https://nodejs.org/>

This automatically includes npm (Node Package Manager)

- Git (for cloning the repository)
 - o Download from: <https://git-scm.com/>
- Modern web browser with JavaScript enabled
 - o Recommended: Chrome, Firefox, Safari, or Edge (latest versions)

Verification

After installation, verify your setup by running these commands in your terminal:

<code>node --version</code>	Should show v16.0.0 or higher
<code>npm --version</code>	Should show 8.0.0 or higher
<code>git --version</code>	Should show git version info

Local Development Setup

Step 1: Clone the Repository

```
git clone https://github.com/andybeecompsci/QLD-Health-Capstone-T302.git
```

Step 2: Install Development Dependencies

Install live-server globally (if not already installed)

```
npm install -g live-server
```

Step 3: Start Development Server

Navigate to project directory (if not already there)

```
cd QLD-Health-Capstone-T302
```

Step 4: Start the development server

```
live-server
```

Step 5: Verify Setup

The development server will automatically open your default browser

Navigate to: <http://localhost:8080/> (or the port shown in terminal)

You should see the Queensland Health Approved Auditors application running

Production Deployment

Deploy to any static web hosting service by uploading all files while maintaining directory structure. Ensure web server serves HTML files correctly. No server-side processing required.

Data Source Configuration

Current Implementation

The application currently fetches data from a self-hosted Excel file due to CORS errors encountered when attempting to access the official Queensland Government URL directly. This is a temporary workaround that can be easily resolved by the internal development team.

```
const excelUrl = "https://api.codetabs.com/v1/proxy?quest=" +  
  encodeURIComponent("https://andersonbee.com/approved-auditor%20(4).xlsx");
```

Production Configuration

To use the official Queensland Government data source, simply update the URL in script.js (line 403) to point directly to the government server:

```
const excelUrl = "https://www.health.qld.gov.au/_data/assets/excel_doc/0016/440044/approved-auditor.xlsx";
```

Core Functionality

Data Processing

- Fetches Excel file from hosted URL
- Parses Excel data using SheetJS library
- Handles merged cells for organization and phone inheritance
- Categorizes auditors by type (Approved, Local Government, QLD Health)

Filtering System

- Text search by name or registration number
- Region filtering by Queensland local government areas
- Auditor type filtering
- Scope filtering (High Risk, Cook Chill, Heat Treatment)

UI Features

- Custom implementation that mimics Queensland Government branding
- The application was built from scratch initially, and while QLD Health provided Storybook and Bootstrap components, we maintained our initial implementation due to time constraints
- The internal development team should be able to implement the official QLD components
- Dynamic card generation for search results
- Randomized display order for fairness
- Expandable region lists for long service areas

Maintenance Notes

- Application is completely client-side
- No database or server-side processing required
- Data updates automatically when Excel file is updated
- Monitor for Excel structure changes that may affect parsing