

# Project Title

PriceScout - a South African price intelligent platform

## Table of Contents

1. Description
2. Tech Assets and URLs
3. Installation
4. Usage
5. Data Sources
6. Troubleshooting
7. Cost Involved
8. Passwords and credentials
9. Contact Information

## Description

The project aims to facilitate the development of a comprehensive platform that enables users to make informed purchasing decisions by aggregating and analyzing retail pricing data from leading South African retailers.

The primary objectives of the project include designing a well-structured database schema, recommending reliable data sources, and offering clear guidance for future development and automation.

By accomplishing these goals, the project establishes a strong foundation that will support GigEfx in advancing the next stages of this platform's development, ultimately addressing the current gap in the South African market for a centralized and transparent price comparison tool.

## Tech Assets and URLs

All assets are uploaded to this public GitHub repository:

<https://github.com/zchen62/BUSA649>

This repository will be available until July 31 2025.

# Installation

## Instructions for Replicating and Accessing the PriceScout Azure SQL Database

You have three ways to replicate and access the PriceScout Azure SQL Database. Please select the option that best suits your needs.

### 1. Deploy Using ARM Template

This method replicates the database schema and server configuration using our exported Azure Resource Manager (ARM) template.

Steps:

1. Initialize Your Azure Subscription
  - a. Ensure you have an [active Azure subscription](#).
  - b. [Login](#) to the Azure Portal.
2. Deploy the ARM Template
  - a. Download the provided ARM template file (Github>Deliverable 1: Database Design>ARM template database.zip).
  - b. In the Azure Portal, search for "Deploy a custom template".
  - c. Click "Build your own template in the editor".
  - d. Paste the contents of the ARM template or upload it.
  - e. Click Save, then Review + Create.
  - f. Fill in any required parameters and click Create.
3. Load sample data using the INSERT Script (Github>Deliverable 1: Database Design>Sample Data Insert Query.pdf).
  - a. Open the [Query Editor (Preview)] in the Azure SQL Database's overview blade, or connect to the database using Azure Data Studio or SQL Server Management Studio (SSMS).
  - b. Run the provided DDL (table creation) and INSERT scripts to populate the database with starter data.

## 2. Connect Directly to the Hosted SQL Instance (Available until July 31, 2025)

You may use the managed instance we are hosting for you through July 31st, 2025.

Steps:

1. Request IP Whitelisting
  - a. Please contact [ziyan.chen@mail.mcgill.ca](mailto:ziyan.chen@mail.mcgill.ca) with your public IP address to request firewall whitelisting.
2. Connect Using a SQL Client
  - a. Use any compatible SQL client (Azure Data Studio, SSMS, DBeaver, etc.).
  - b. Use the credentials and database name found in the Passwords and credentials section.

## 3. Manual Database Creation

If you prefer, you can replicate the environment on any SQL Server (local, cloud, or other provider):

Steps:

1. Create a new SQL database on your chosen server.
  - a. Run the provided DDL script to create the tables and structure (Github>Deliverable 1: Database Design>DDL for database creation.pdf).
  - b. Run the INSERT script to populate initial/test data (Github>Deliverable 1: Database Design>Sample Data Insert Query.pdf).

## Usage

This project provides foundational prototype and guidance for building, extending, and integrating the PriceScout platform. Below are ways you can leverage the included resources to accelerate your development and data integration tasks.

### 1. Continue Development with a Ready-to-Use Database

You are provided with detailed instructions and scripts to easily create or replicate the designed database on Azure SQL or on any compatible SQL Server instance.

Feel free to use this database setup as a foundation for your own feature development, customization, and testing. You can extend tables, add new relations, or load additional sample data as needed in the future.

## **2. Informed Development Sequence Using Data Source Recommendations**

The Local Market Research section offers prioritized recommendations for different data sources based on data quality and integration complexity.

Use these recommendations to:

- Plan your roadmap: Start integrating sources that are easiest and most reliable first.
- Allocate resources efficiently: Focus on sources with high impact and low difficulty.
- Anticipate challenges: Understand which data sources may require more work, so you can schedule them appropriately in your development cycle.

This helps you make informed decisions about the preferred order to tackle data integrations and maximizes your development momentum.

## **3. Accelerate Data Ingestion Using Data Retrieval & Mapping Guide**

The Data Retrieval Guide maps retailer platform data fields directly to your database schema.

You can use this guide to:

- Streamline web scraping: See exactly where each piece of retailer data should go in the database.
- Minimize onboarding time for new developers: Anyone joining the project will quickly understand the correlation between raw data and storage structure.
- Increase accuracy and consistency: Reduce errors by following clear, field-by-field mapping during integration.
- Facilitate automation: Reference the guide during scraper or ETL development to check mappings.

## Data Sources

The project utilized a combination of reputable and relevant data sources to ensure comprehensive coverage and accuracy. Key sources include:

Retailer Websites:

Company and product information was gathered directly from leading South African retail platforms, including Pick n Pay, Woolworths, Makro, Takealot.com and Checkers.

Statista Reports:

Market analysis and industry statistics were supported by data from Statista reports (2023, 2025), providing insights into smartphone user trends and retail market growth in South Africa.

These sources ensured that both primary, real-time retail data and contextual industry statistics were integrated into the project's analysis.

## Troubleshooting

If you encounter errors when connecting to the Azure SQL Database (using either the ARM template deployment or direct instance connection):

### 1. Check the Error Message

Read the error message carefully—it often indicates the exact issue, such as:

IP not allowed by firewall

Incorrect username or password

Database server unreachable

TLS/Encryption issues

### 2. Use the Azure Portal's Support + Troubleshooting Section

Azure provides built-in tools and guidance to help you resolve connection problems.

Step-by-Step:

1. Go to the Azure Portal <https://portal.azure.com/>
2. Navigate to Your SQL Database or SQL Server Resource
  - a. Search for your SQL database or server in the top search bar.
  - b. Click to open the resource.
  - c. Locate the "Support + Troubleshooting" Section
  - d. In the left-hand menu, look for "Support + troubleshooting".
  - e. Under this section, click on "Troubleshoot connectivity".
  - f. Follow the On-Screen Troubleshooting Tools

## Costs involved

The Azure SQL database used for the prototype leveraged a limited-time offer that included 100,000 vCore seconds, 32GB of data, and 32GB of backup storage free per month for the lifetime of the subscription (up to ten free databases per subscription). As a result, no costs were incurred during this phase, given the modest usage level required for the prototype. Additionally, the database is scheduled to be taken offline on July 31, further ensuring that no ongoing or unexpected costs will be incurred.

## Passwords and credentials

### Connection String:

1. ODBC:

```
Driver={ODBC Driver 18 for SQL  
Server};Server=tcp:priceccout.database.windows.net,1433;Database=PriceScout;Uid=vh  
wjebgwst;Pwd= 8viLo6wKD6Bht$Te;Encrypt=yes;TrustServerCertificate=no;Connection  
Timeout=30;
```

2. php:

```
<?php
```

// PHP Data Objects(PDO) Sample Code:

```
try {  
    $conn = new PDO("sqlsrv:server = tcp:priceccout.database.windows.net,1433; Database  
    = PriceScout", "vhwjebgwst", "8viLo6wKD6Bht$Te");  
    $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);  
}  
catch (PDOException $e) {  
    print("Error connecting to SQL Server.");  
    die(print_r($e));  
}
```

// SQL Server Extension Sample Code:

```
$connectionInfo = array("UID" => "vhwjebgwst", "pwd" => "8viLo6wKD6Bht$Te",  
"Database" => "PriceScout", "LoginTimeout" => 30, "Encrypt" => 1,  
"TrustServerCertificate" => 0);  
$serverName = "tcp:priceccout.database.windows.net,1433";  
$conn = sqlsrv_connect($serverName, $connectionInfo);  
?>
```

## Contact Information

For any questions or concerns, please contact:

- **Name:** Ziyang Chen
- **Email:** ziyang.chen@mail.mcgill.ca
- **GitHub:** <https://github.com/zchen62>