# SEED
## INTERACTIVE

V 1.0.0 (Self-Hosted)
Last Updated:  Feb 21th 2022

# Introduction

       This guide will provide a basic walkthrough of a self hosting setup of the butchery application using windows server with IIS 7+. Some additional modules will need to be installed IISNODE, and the REWRITE module, This allows node to run on IIS and the REWRITE module allows for IIS to rewrite the requested URL paths to the butchery application correctly (SPA APPLICATION require rewrite module). The basic config for these modules will be contained in the application package. This application also uses a Google Service account to POST / APPEND scores to excel sheets, so that the instructors can easily view students results. For this we will be creating a service account, an excel document and we will be adding the ID's to the applications environment variables.

# Requirements

- Windows server environment
- IIS 7+
- Google Account (Hosting online spreadsheets + use of Sheets API)
- Google Console Project with Sheets API Enabled (setup in guide)
- Project is downloaded and unzipped ready to host:
  https://previewseed.s3.us-east-2.amazonaws.com/next-iis.zip

# Installing Node + IIS Modules

You will **need to install these packages** for IIS to work with the application.

- Installing Node: https://nodejs.org/en/download/
  - Install the one according to you bitness, 64bit or 32bit

## Downloads

Latest LTS Version: **16.14.0** (includes npm 8.3.1)

Download the Node.js source code or a pre-built installer for your platform, and start developing today.

| LTS Recommended For Most Users | | Current Latest Features | |
|---|---|---|---|
| Windows Installer | macOS Installer | | Source Code |
| node-v16.14.0-x64.msi | node-v16.14.0.pkg | | node-v16.14.0.tar.gz |

| | | |
|---|---|---|
| Windows Installer (.msi) | 32-bit | 64-bit |
| Windows Binary (.zip) | 32-bit | 64-bit |
| macOS Installer (.pkg) | 64-bit / ARM64 | |
| macOS Binary (.tar.gz) | 64-bit | ARM64 |
| Linux Binaries (x64) | 64-bit | |
| Linux Binaries (ARM) | ARMv7 | ARMv8 |
| Source Code | node-v16.14.0.tar.gz | |

**Installing URL Rewrite for IIS:**

https://www.iis.net/downloads/microsoft/url-rewrite

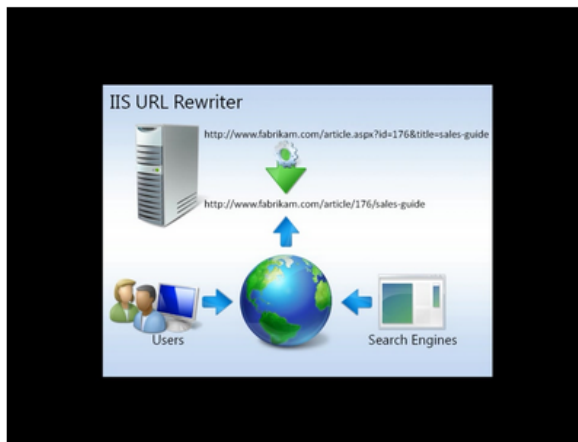## URL Rewrite

This is a Microsoft Supported Download  |  Works With: IIS 7, IIS 7.5, IIS 8, IIS 8.5, IIS 10

Install this extension

or view additional downloads

## Overview

IIS URL Rewrite 2.1 enables Web administrators to create powerful rules to implement URLs that are easier for users to remember and easier for search engines to find. By using rule templates, rewrite maps, .NET providers, and other functionality integrated into IIS Manager, Web administrators can easily set up rules to define URL rewriting behavior based on HTTP headers, HTTP response or request headers, IIS server variables, and even complex programmatic rules. In addition, Web administrators can perform redirects, send custom responses, or stop HTTP requests based on the logic expressed in the rewrite rules.

- **Installing IIS Node:** https://github.com/Azure/iisnode

Scroll down on the gitpage till you see install instructions, click x86 or x64 for the installer. (Run the installer)

# Create Site in IIS

This section assumes you have an IIS setup for your network and know what settings you need for your specific network. This section will just cover selecting the application in the ADD SITE modal, and showing the application started.
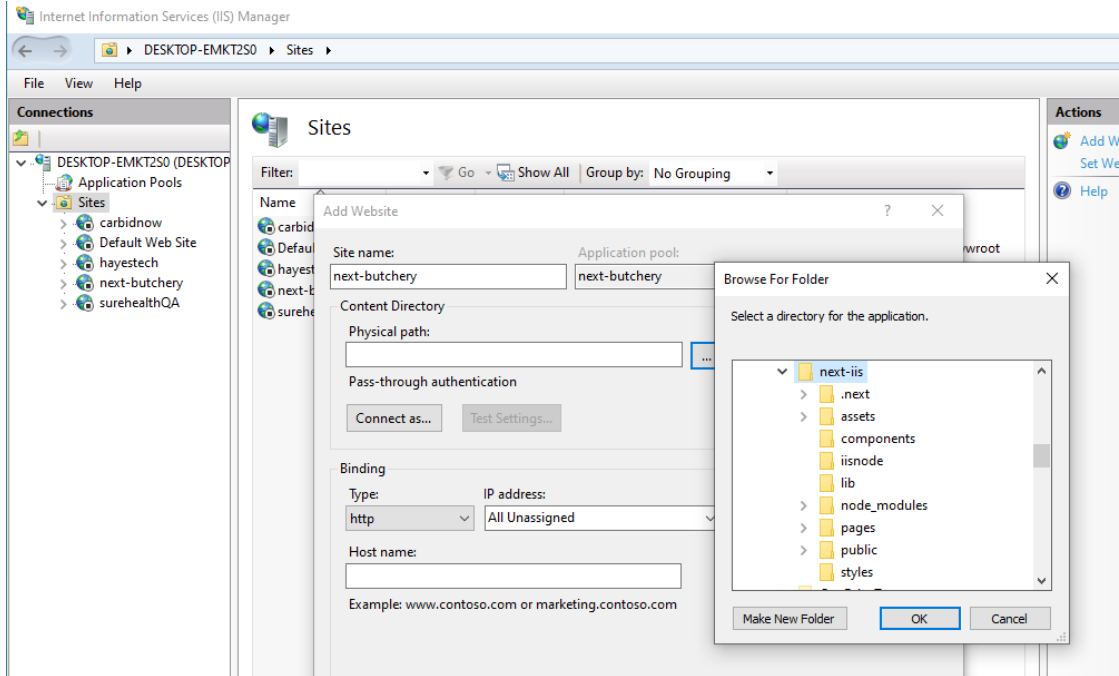
## Opening the Add Site:

Right click on sites in the IIS panel.
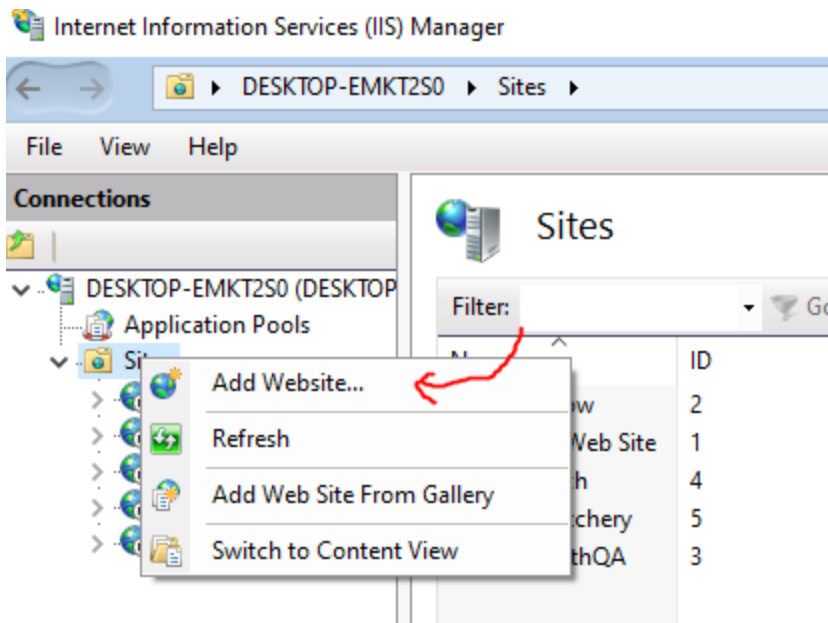


This will bring up the add site prompt, Where you can insert the values you will need for your network. In the case of the demo, I will be running locally on port 3000.

## Selecting the project:
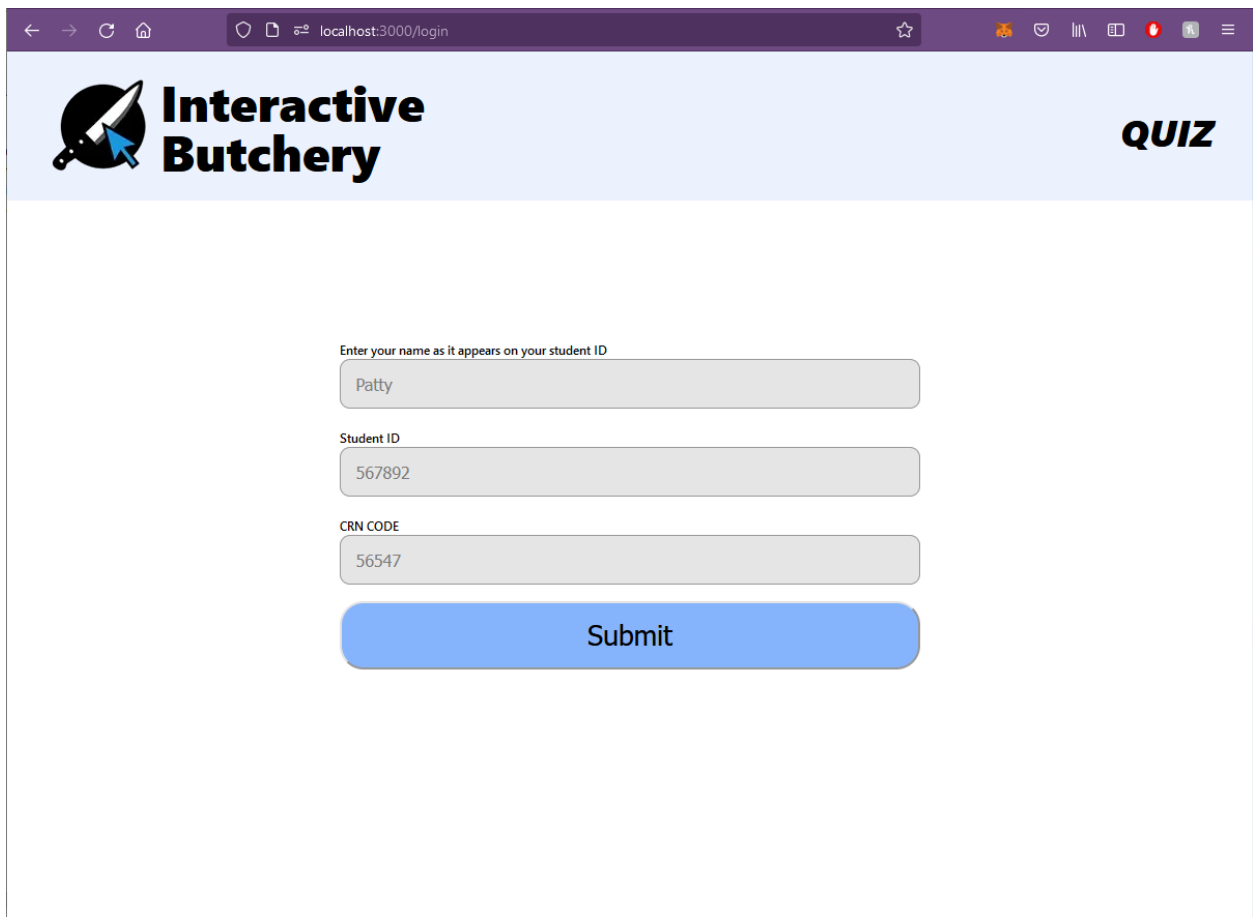
On the physical path of the add site modal, select the root of the project, In this case next-iis.



After completing this add site modal, the website should be started on the port / network you selected. My case, localhost:3000, You can browse to the site by right clicking the site you created and selecting Browse
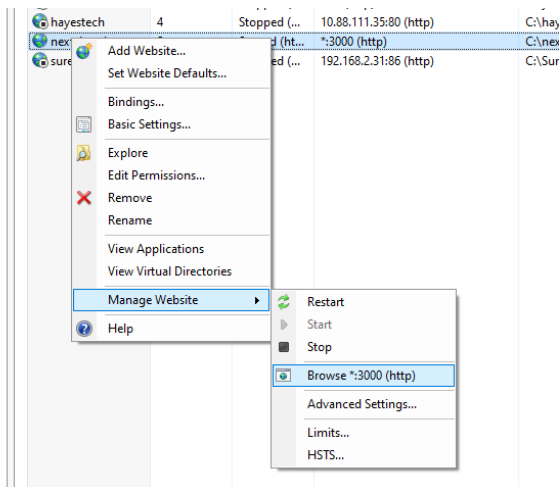
You should be presented with the login page.



## Setup Google Service Account and Sheets

The butchery application reports logins and quiz results to 2 Google Sheets documents hosted online. In this section we will be setting up the google service account, attaching the service account credentials to the project, Copying the google sheet format to our account and adding those generated Ids to the project as well.

## Creating the Service Account

Navigate to [https://console.cloud.google.com](https://console.cloud.google.com) with the account you will be using to setup the service account and share the hosted documents with. You should be prompted to create a new project, if not, create a new project. (you may have a different view depending on your account level, just need to create a new project, Here is my dashboard view for organization account)
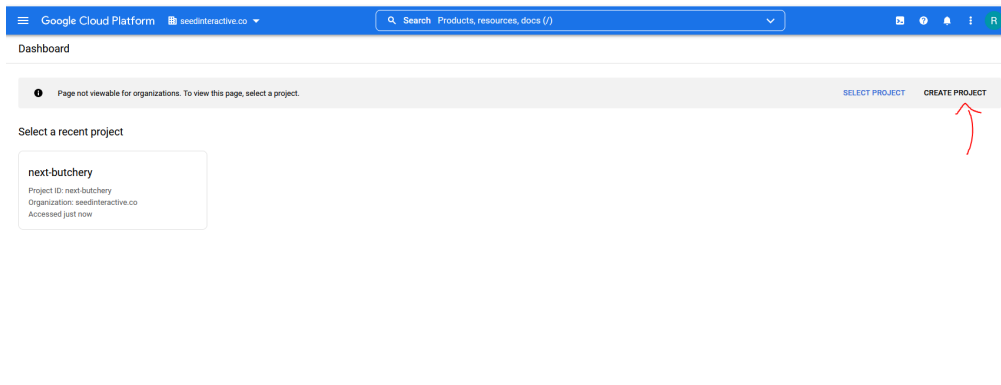




Name the project however you prefer and press create, You should be presented with a dashboard,

**Select, Go to Project Settings -> Service accounts**

**Project info** ⋮

Project name
test-next-doc

Project number
239554472665

Project ID
test-next-doc

**ADD PEOPLE TO THIS PROJECT**

→ Go to project settings

**API APIs**

Requests (requests/sec)

⚠ No data is availabl

8:30            8:45

→ Go to APIs overview

**Resources** ⋮

BigQuery
Data warehouse/analytics

SQL
Managed MySQL, PostgreSQL, SQL Server

Compute Engine
VMs, GPUs, TPUs, Disks

Storage
Multi-class multi-region object storage

Cloud Functions
Event-driven serverless functions

IAM

Identity & Organization

Policy Troubleshooter

Policy Analyzer

Organization Policies

Service Accounts

Workload Identity Federation

Labels

Tags

Settings

Privacy & Security

Identity-Aware Proxy

Roles

## Service accounts for project "test-next-doc"

A service account represents a Google Cloud service identity, such as code

Organization policies can be used to secure service accounts and block ris

≡ Filter    Enter property name or value

| ☐ | Email | Status | Name ↑ | Descriptio |
|---|-------|--------|--------|------------|

No rows to display

Select Create Service account (top bar), Here is what I filled in for this example

## Create service account

**1** **Service account details**

Service account name *

test service account

Display name for this service account

Service account ID *

test-service-account          @test-next-doc.iam.gserviceaccount.com  ✕  ↻

Service account description

Testing service account for example document

Describe what this service account will do

CREATE AND CONTINUE

**2** **Grant this service account access to project** (optional)

**3** **Grant users access to this service account** (optional)

DONE     CANCEL

**Click Done,**

Now, we will need to get the service key for our newly created service account,

On the newly created service account,  Click **Manage Details -> Keys**



On the keys screen we're going to create key of JSON type
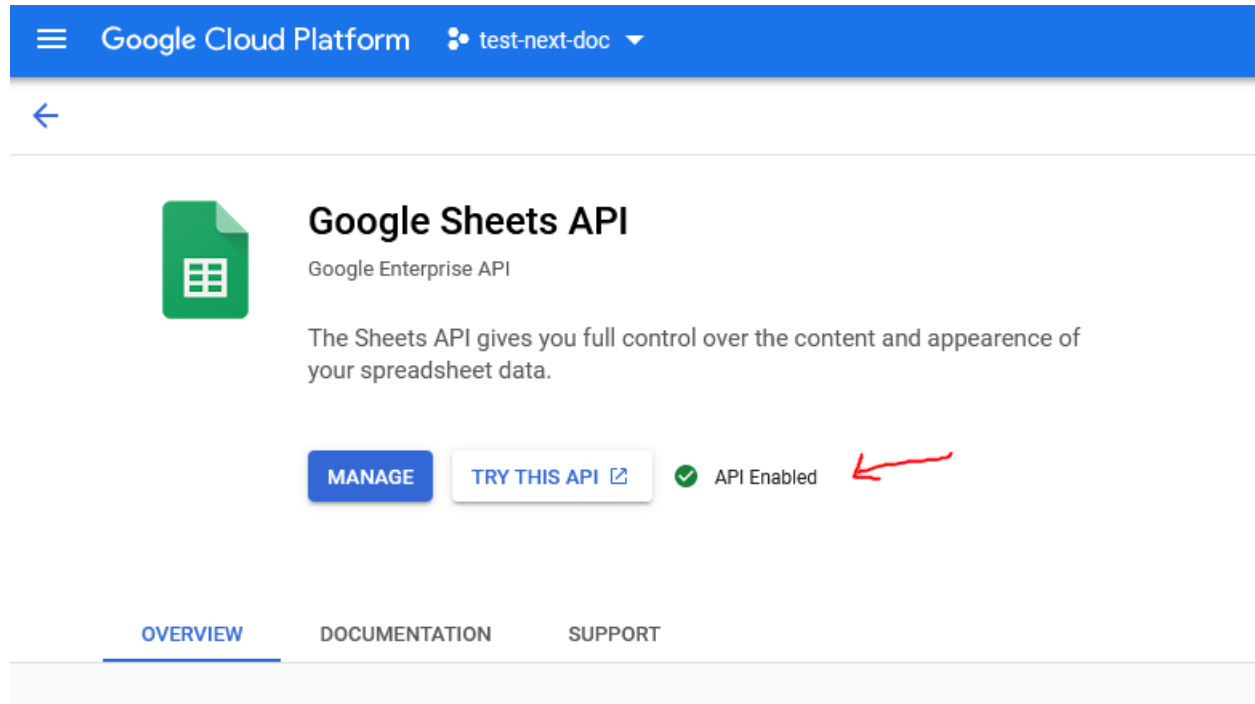


This will create a JSON file and download it to your computer, Keep this for later as it has the private key and email of this service account, We will need to input it into the project environment variables.

**Lastly enable sheets API on the project you created**
https://console.developers.google.com/apis/library/sheets.googleapis.com



## Copying Login & Results Sheet

This section will have you copy 2 template sheets to your google drive with your google account, These sheets are the login tracker sheet and the quiz results. Once copied we will take the generated ID and add them to the project's environment variables.
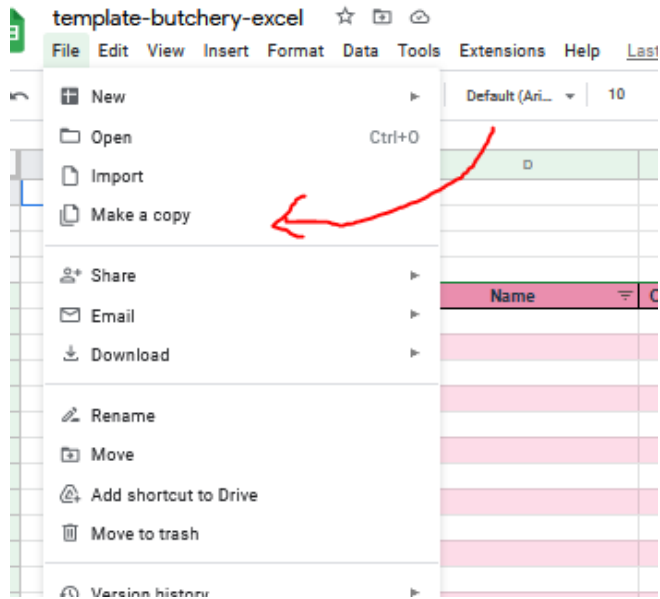
Navigate to the result sheet and **File -> Make a Copy**

**RESULT SHEET**

https://docs.google.com/spreadsheets/d/1BEvpO5J6H6vDXG1E_EH8fYfJsqAMZVVJRsSHdE3Mp1o/edit?usp=sharing

**Login Sheet**
https://docs.google.com/spreadsheets/d/1Z3auRSpUXhxeQx_6a73uSzzrAHZtTTaTV6ou9S8HRFQ/edit?usp=sharing

This should copy the spreadsheet to your account,

Once the files are in your account you want to share access with the service account you created, Click on the share button and paste in the email you see within the JSON key you downloaded earlier.
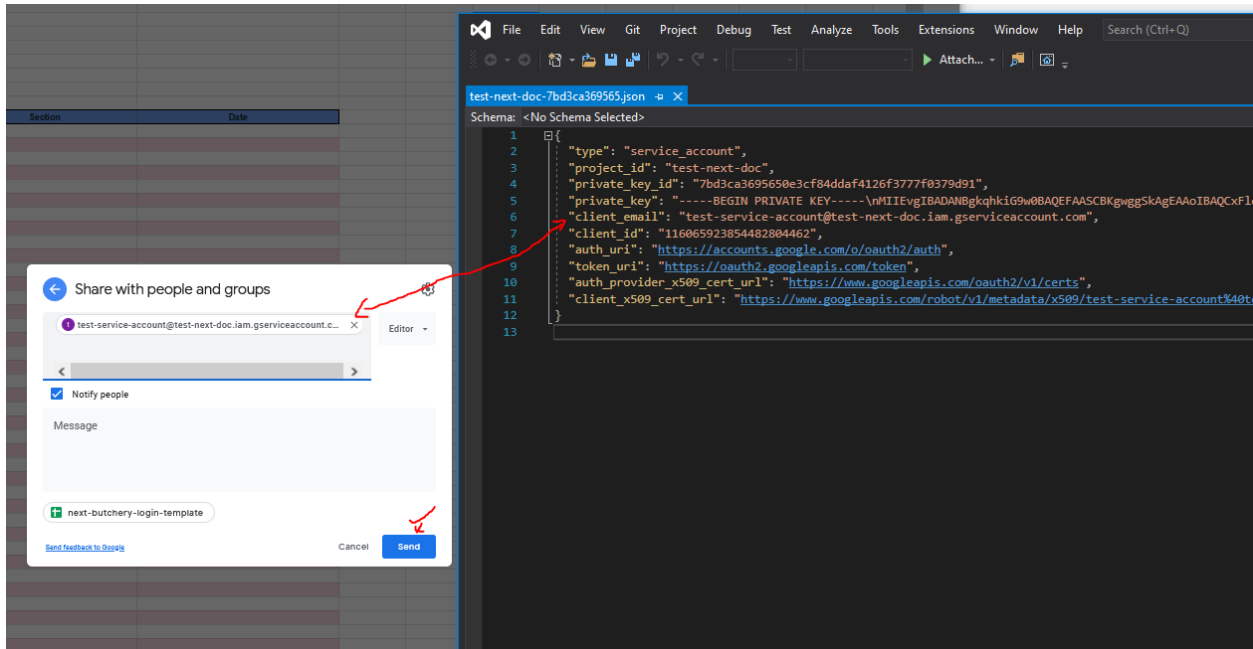


Copy the URL from the web address of the new copied spreadsheet and take the ID from the center of that URL,
Example case: (ID is highlighted)
ttps://docs.google.com/spreadsheets/d/1BEvpO5J6H6vDXG1E_EH8fYfJsqAMZVVJRsSHdE3Mp1o/edit#gid=0

Copy these ID to a file in the project called .env.local





GOOGLE COMPLETE SHEET= Results Sheet ID

GOOGLE LOGIN SHEET = Login Sheet ID

**MAKE SURE TO DO THIS FOR BOTH SHEETS**

**KEEP THIS OPEN FOR THE NEXT SECTION**

## Adding Service Account Key to the project.

Open the JSON key saved to your computer earlier when creating the service keys for Service accounts,

Mine looks like this:



Copy the Private key & Email into the **.env.local** in this format

```
env.local
    GOOGLE_SERVICE_PKEY="-----BEGIN PRIVATE KEY-----\nMIIEvgIBADANBgkqhkiG9w0BAQEFAASCBKgwggSkAgEAAoIBAQCxFlo
    GOOGLE_SERVICE_EMAIL=test-service-account@test-next-doc.iam.gserviceaccount.com
    GOOGLE_LOGIN_SHEET=1Z3auRSpUXhxeQx_6a73uSzzrAHZtTTaTV6ou9S8HRFQ
    GOOGLE_COMPLETE_SHEET=1BEvpO5J6H6vDXG1E_EH8fYfJsqAMZVVJRsSHdE3Mp1o
    SECRET_COOKIE_PASSWORD=GjB7TFTxBdxv6vYhZrHAQABAoGAX58GJBAMMERAQABAoGAX58GJHTSpfmr0ubZ6m
```

**NOTE: KEY keeps the quotes (") and email does not**

Save the new environment variables ( file->save ) and restart the server using IIS.
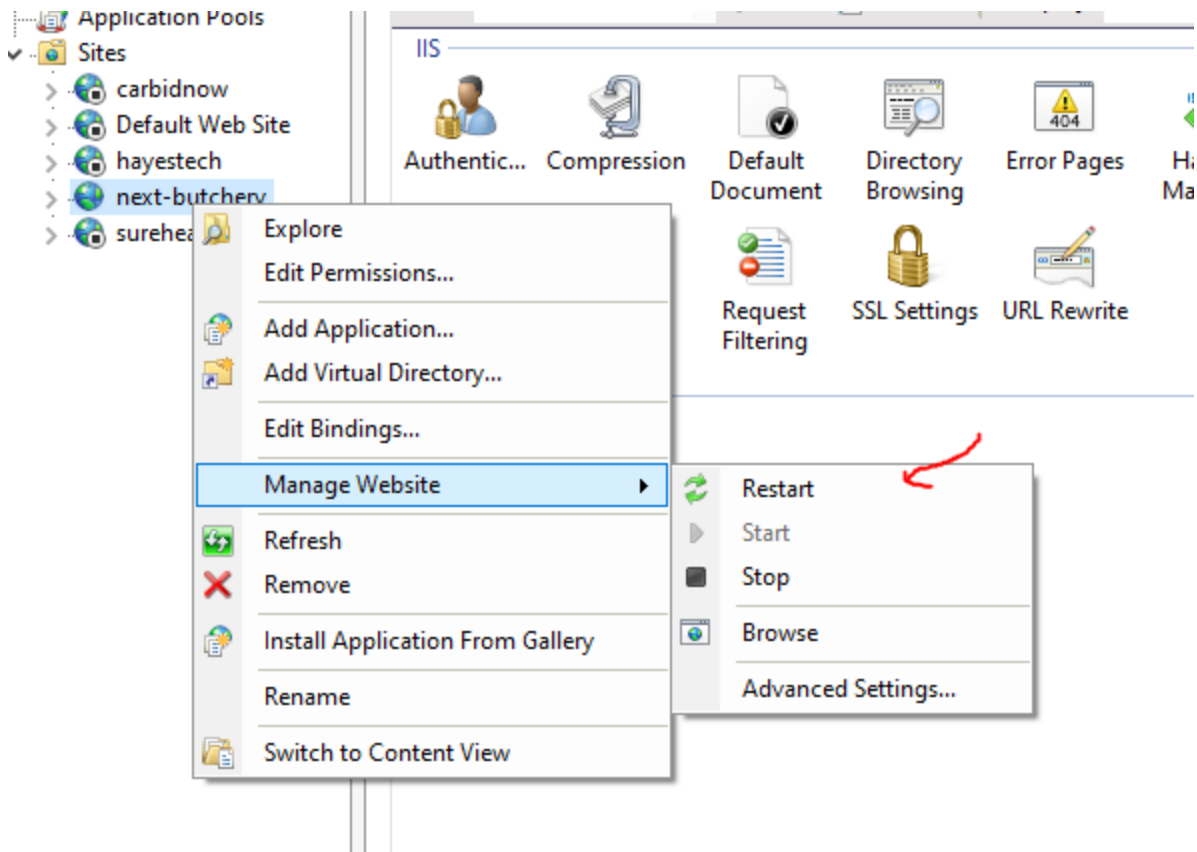


Upon restart you project should be viewable and when you login the info should be viewable in the login sheet + results on the login spreadsheets you created.
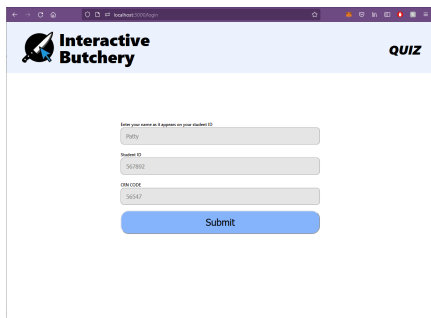
# Testing

Here are some training module paths you can test, to go to the quiz you will have to login but the button is bottom center


/3DApp/template.html?meat=Beef

/3DApp/Chicken/splashChicken.html


Also check that your logging the results of the Login page and Results page on the Spreadsheets ID you added to the .env.local file - If not ensure you restarted the server and check if it logged them on the location you copied it from