

Butchery

Fundamentals

[Demo Version of Butchery Application](#)

[Demo Version of Butchery Application Test Result](#)

[Demo Version of Butchery Application Logged Student](#)

Project Overview and Upload Guide

In this documentation, we provide a guide regarding what you'll require for uploading the application to [Vercel](#). This application is a static next JS site. The project files can be found at [Butchery Fundamentals](#). For additional assistance, you can contact the following people

Project Owner: [Daniel Stopnicki](#)

Developer: [Mas Tsukada](#)

Table of Content

Project Overview and Upload Guide	0
Table of Content	1
Required Skills, Items, and Experience for project setup and maintenance	2
Pre-Upload steps	3
1. Initial Setup	3
2. How to create a project on Google Cloud services	3
3. How to create a Google service account for API Google sheet calls	5
UPLOADING TO VERCEL	8
Enviroment Variables table	9

Required Skills, Items, and Experience for project setup and maintenance

- Experience with GitHub
- Experience with Node JS
- Experience with Web Application Frameworks such as
 - Angular
 - React
 - Next.js
- (Not necessary but would be a bonus) Experience with [Google Cloud Services](#)
- Requires an account with [GitHub](#)
- Requires a [Gmail](#) Account
- Requires a [Vercel](#) Account (can be created with GitHub account)

*** Recommended user for Setup and maintenance is an IT professional ***


Pre-Upload steps

In this section, we will cover the initial steps needed prior to uploading the project to **VERCEL**. Here are the following topics that will be covered in this section

- How to create a project on Google Cloud services
- How to create a Google service account for API Google sheet calls

These steps can also be found under the [updating the Google Sheets service call](#) on GitHub

Prior to doing these steps create a copy of these Google Sheets as the information regarding the results of tests and logged in users will be stored here

Login template Sheet	 next-butchery-login-templ...
Results template sheet	 template-butchery-excel

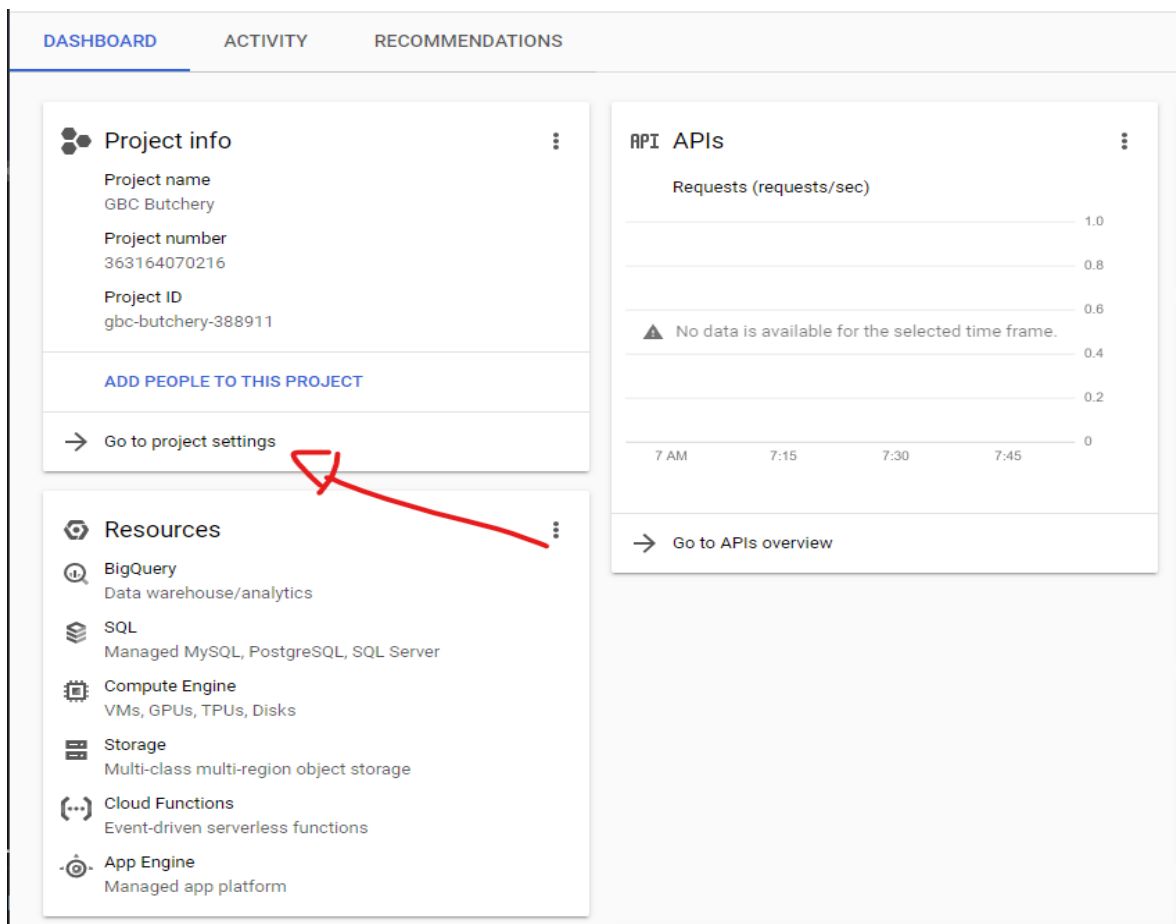
1. Initial Setup

1. Fork the [GitHub project](#) to your Account

2. How to create a project on Google Cloud services

1. Go to Google cloud services

2. In the top left corner where it says **Select a project** press and select **New Project** in the new window
3. After naming the project press the **Create** button
 - once pressed a process will start where it creates the portal platform for the project (press the **go to dashboard** once the process is finished)
4. In the top left inside of the dashboard section press the **Go to Project settings** button



3. How to create a Google service account for API Google sheet calls

1. Once inside the project setting section **press Service Accounts** -> **CREATE SERVICE ACCOUNT (on top bar)**

The screenshot shows the Google Cloud IAM & Admin console for project "GBC Butchery". The left sidebar is expanded to "IAM & Admin", and the "Service Accounts" option is highlighted with a red arrow and the number "1.". In the main content area, the "+ CREATE SERVICE ACCOUNT" button is highlighted with a red arrow and the number "2.". Below the button, there is a table with columns: Email, Status, Name, Description, Key ID, and Key creation date. The table is currently empty, showing "No rows to display".

2. Name the service account name whatever you'd like and **press done**
Once the account has been created press the **Actions** button and then **Manage Details**

<input type="checkbox"/>	Email	Status	Name ↑	Description	Key ID	Key creation date	OAuth 2 Client ID ⓘ	Actions
<input type="checkbox"/>	gbc-butcher-api@gbc-butcher-388911.iam.gserviceaccount.com	✓	GBC butchery api		No keys		104332376376218586953	<ul style="list-style-type: none"> 1. Manage details 2. Manage permissions Manage keys View metrics View logs Disable Delete

3. In this new screen press **Keys** and **Add key -> Create new key**

DETAILS PERMISSIONS KEYS METRICS LOGS

Keys

⚠ Service account keys could pose a security risk if compromised. We recommend you avoid downgrading service accounts on Google Cloud [here](#).

Add a new key pair or upload a public key certificate from an existing key pair.

Block service account key creation using [organization policies](#).
[Learn more about setting organization policies for service accounts](#)

ADD KEY ▾

Key creation date
Key expiration date

Create new key

Upload existing key

4. Select the **JSON** option in the window that pops up and press **create** doing this step will then save a .JSON file to your computer

(do not destroy the .json file as we will be needing it for future steps)

5. Open this .JSON file with some text editor of your choice and copy the value that is associated with the key **client_email**

EXAMPLE:

"client_email": gbc-butchery-api@gbc-butchery-388911.iam.gserviceaccount.com",

KEY	VALUE
"client_email":	gbc-butchery-api@gbc-butchery-388911.iam.gserviceaccount.com",

6. In the google sheets that you made a copy of earlier in the steps press the **share** the button and paste the email that you copied from the previous step

Example	https://github.com/DDSkunkworks/ButcheryFundamentals/blob/main/readmeassets/sharegmail.gif
---------	---

7. Enable the use of the [google sheets api](#) to allow the email write access to the sheets

UPLOADING TO VERCEL

In this section, we will cover the initial steps on how to upload the project to Vercel

Please note that the steps from the pre-upload need to be completed before moving on to this step

1. Fork this project into a GitHub account that will be associated with Vercel
2. Go to Vercel and log in or signup

(it is recommended to select the GitHub option, use the GitHub account from step 1 of Uploading to Vercel)
3. once on the dashboard press the **add new** button in the top right-hand corner
4. select the **project** option
5. Press the dropdown option of **select a Git Namespace** and select **Add GitHub account**
 - a pop will appear and ask which account you'd like Vercel to connect too
 - select the option where you forked this project too
6. once on the "almost done" page press the **Environment Variables** section
7. Copy the id of both google sheets from the url and paste the values onto vercel as values to the same variable in **Environment Variables table**

EXAMPLE:

https://docs.google.com/spreadsheets/d/11SijW8y4XHpfQGWHfL4sOjC9yIZ1slH5cKcrUBWlvWg/edit#gid=0

id :	11SijW8y4XHpfQGWHfL4sOjC9yIZ1slH5cKcrUBWlvWg
------	--

Environment Variables table

using [Base code 64](#) copy all of the contents from that .json file (**How to create a Google service account for API Google sheet calls -> step 4**) and paste them into the encoder section of the web application, once pasted press the encode button and save the results to the **GOOGLE_SERVICE_KEY**

ENVIRONMENT VARIABLE VALUE

Environment variable name	value
SECRET_COOKIE_PASSWORD	GjB7TFTxBdxv6vYhZrHAQABAoGAX58GJHTSpfmr0ubZ6m
GOOGLE_COMPLETE_SHEET	ID FROM RESULTS GOOGLE_SHEET
GOOGLE_LOGIN_SHEET	ID FROM LOGIN GOOGLE_SHEET
GOOGLE_SERVICE_KEY	Base 64 ENCODED STRING

DO NOT CHANGE THE SECRET_COOKIE_PASSWORD

The Environment variable's names on Vercel should be the exact same as the ones in the table

