



Mercy Clinic's Volunteer Management System

Developer Manual

Version 1.3

Revision History

Date	Version	Description	Author
4/13/20	1.0	Initial Draft of Developer Manual Sections: 1.0,1.1,2.0,2.1,2.1.1,2.1.2,2.1.3,2.2,2.2.1,2.2.2,2.2.3,2.3,2.3.1,2.3.2,2.3.3, 3.1	Kenzie Clarke
4/19/20	1.1	Added sections: 2.4, 4.1, 4.2 Added pictures: 3.1	Warren Riley
4/19/20	1.2	Added sections: 5.1,5.2,5.3	Kenzie Clarke
4/20/20	1.3	Added sections: 5.4, 6.1, 6.2	Kenzie Clarke

Table of Contents

Developer Manual	1
Introduction	4
Purpose	4
Project Overview	4
Development Environment Setup	4
Nodejs	4
Version Check	4
Downloading Nodejs	4
Installing Nodejs	4
MySQL Workbench	5
MySQL Version Check	5
Downloading MySQL Workbench	5
Installing MySQL Workbench	5
IDE	5
Visual Studio Code	5
Github Desktop	5
Downloading Github Desktop	5
Installing Github Desktop	6
Connecting to Online Github Account	6
Project Setup	11
Loading the Project	11
Importing Dependencies	11
Running the Program	11
Running on Localhost	11
Viewing the ClearDB database on MySQL Workbench	11
Deployment and Heroku Dyno Configuration	13
Setting up Heroku Dyno	13
Deploying Application on Heroku	14
Connecting ClearDB to dyno	16
Viewing Heroku Dyno Metrics	17
Further Resources	18
SSL Certificate Monitoring	18
Super Admin Account	18

1. Introduction

1.1 Purpose

The purpose of the Developer's Manual is to provide a comprehensive guide for developing and maintaining the Mercy Clinic Volunteer System. It describes how to install needed software, how to access the database, and how to manage the Heroku application. Anyone wishing to perform maintenance or add new features to the project should reference this manual for help.

1.2 Project Overview

2. Development Environment Setup

2.1 Nodejs

The application is developed using a Nodejs/Express/HTML/CSS stack. Nodejs is the main programming language for the project.

2.1.1 Version Check

To see if Nodejs is already installed on your machine, enter the following command into the terminal:

```
$ node -v
```

This will print the current version of Nodejs you have, if any. You will need Nodejs v12.15.0 or higher.

2.1.2 Downloading Nodejs

If you do not have Nodejs on your machine, you will need to install it. First, navigate to the official Nodejs download page: <https://nodejs.org/en/download/>. Select the green button for your appropriate operating system : Windows installer, MacOS installer, or source code (for Linux).

2.1.3 Installing Nodejs

Navigate to the downloads folder of your machine and double click the installer to begin installation. If you are on a linux machine, you need to move the .tar file to your destination directory.

Next, unpack the java tarball (use the correct file name for your installation):

```
$ sudo tar xzvf node-v12.15.0.tar.gz
```

At this point you can delete the .tar file if you would like. The installation is complete.

2.2 MySQL Workbench

You will need MySQL Workbench to view and manipulate the ClearDB Database information for the Mercy VMS.

2.2.1 MySQL Version Check

To see if MySQL Workbench is already installed on your machine, type 'MySQL' into your search bar or finder of your operating system. The application 'MySQL Workbench 8.0' or higher should appear in the results.

2.2.2 Downloading MySQL Workbench

If you do not have MySQL Workbench, you will need to install it. First, navigate to the page <https://dev.mysql.com/downloads/workbench/>. Select your operating system in the drop down, and click the button to download the SQL Installer. (Note that if you do not have MySQL at all, you will also need to download this installer for the community version: <https://dev.mysql.com/downloads/mysql/>.)

2.2.3 Installing MySQL Workbench

Navigate to your downloads folder and double click on the installer file you downloaded. The installer should take you through the steps to download MySQL

2.3 IDE

For this project, any IDE can be used as long as it supports the stack mentioned in section 2.0. Our team chose to use Visual Studio Code - it's download and use is entirely free.

2.3.1 Visual Studio Code

To download Visual Studio Code, navigate to the official download page: . Select ... Follow the on-screen prompts to set up the IDE.

For the remainder of this manual, we will assume you are using VSCode. If you are using a different IDE, the steps to import and run the project may vary slightly. Be sure to consult your IDE's official documentation for assistance.

2.4 Github Desktop

2.4.1 Downloading Github Desktop

To download the Github Desktop version you first need to visit <https://desktop.github.com/> and choose the download for your type of computer.

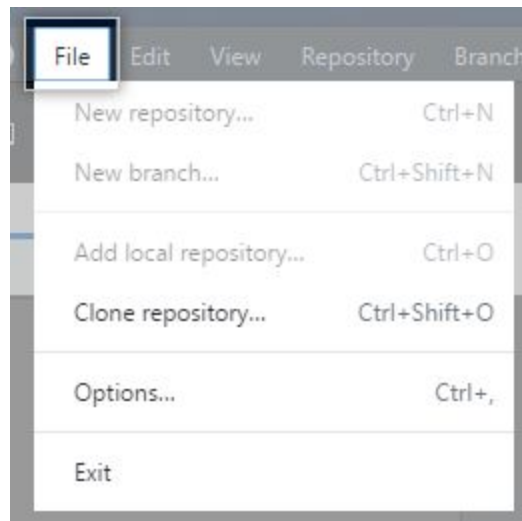
2.4.2 Installing Github Desktop

In your computer's Downloads folder, double-click GitHub Desktop.

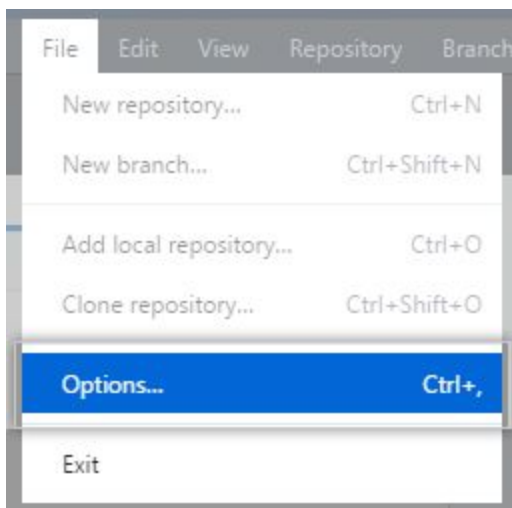
In the pop-up window, click Install.

2.4.3 Connecting to Online Github Account

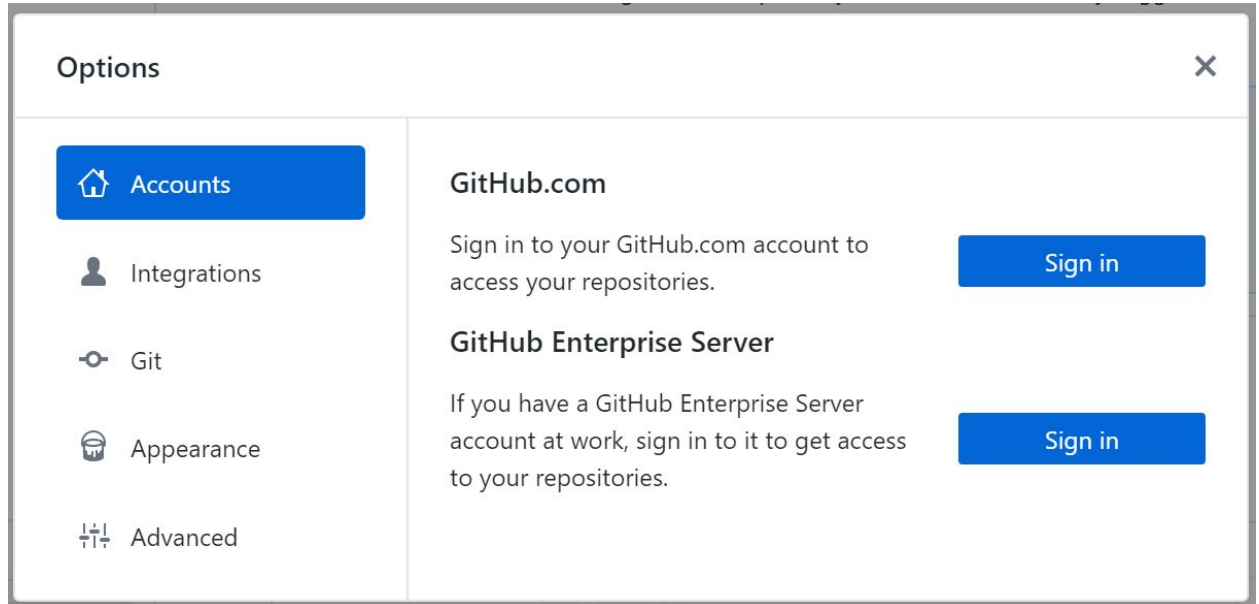
In the upper-left corner of the window, select the File menu.



Click Options.



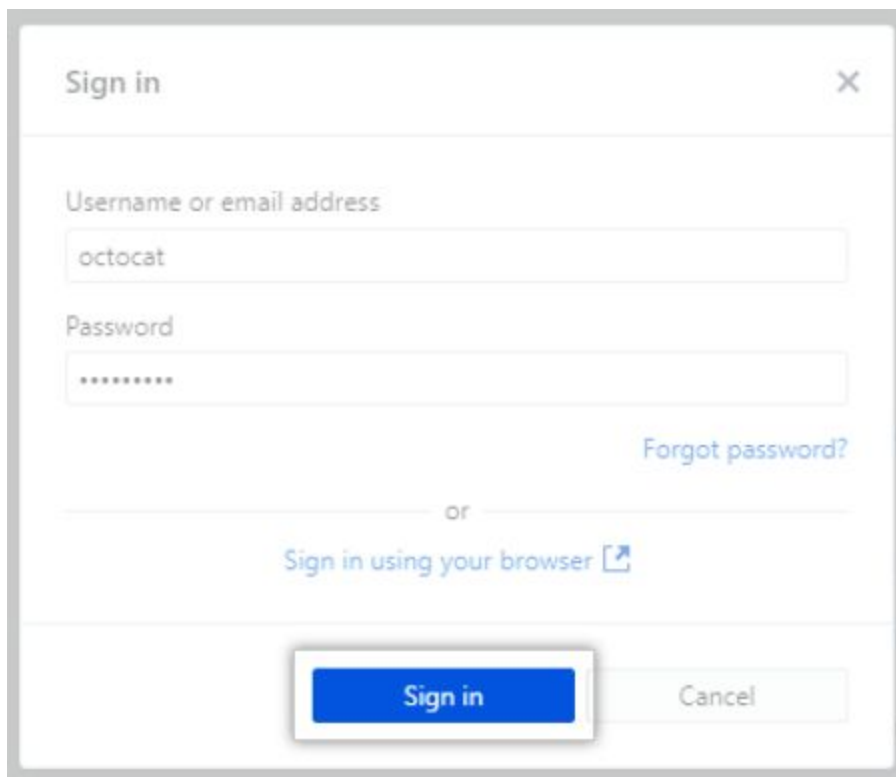
In the Options window, select Accounts.



In the Accounts pane:

To authenticate to GitHub, under "GitHub.com" click Sign In.

To add a GitHub account, type your GitHub.com credentials and click Sign in.

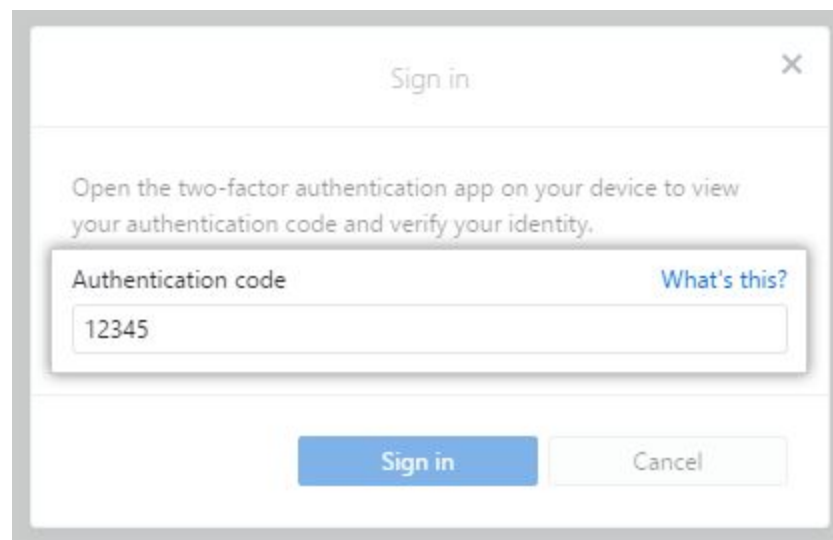


If you have configured two-factor authentication for GitHub or GitHub Enterprise, do one of the following:

If you set up 2FA via SMS, retrieve your 2FA code from an SMS message.

If you set up 2FA with a TOTP application, generate a 2FA code.

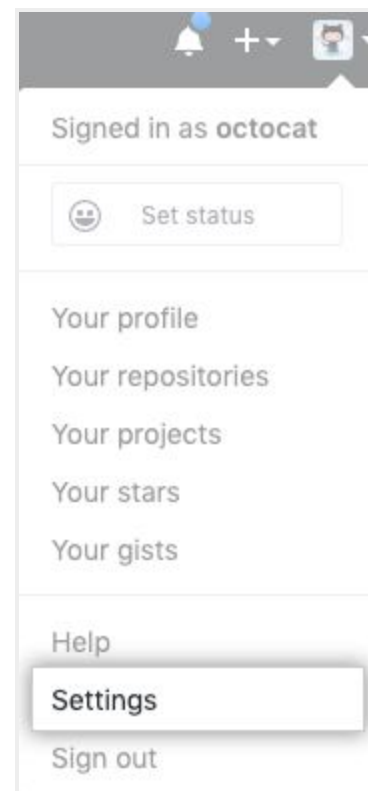
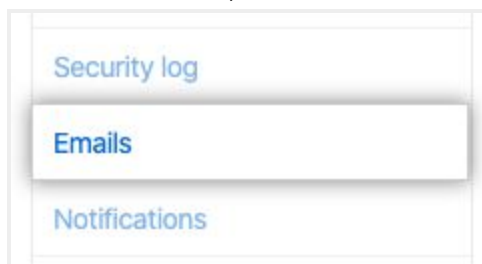
Return to GitHub Desktop. In the prompt, enter your 2FA code, then click Sign in.



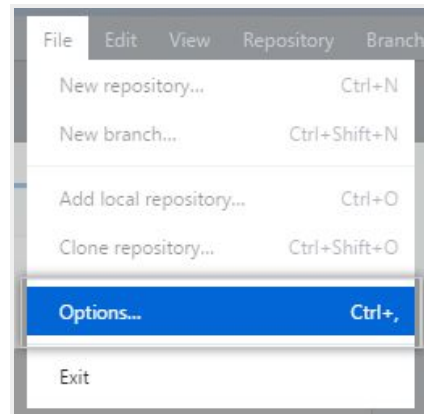
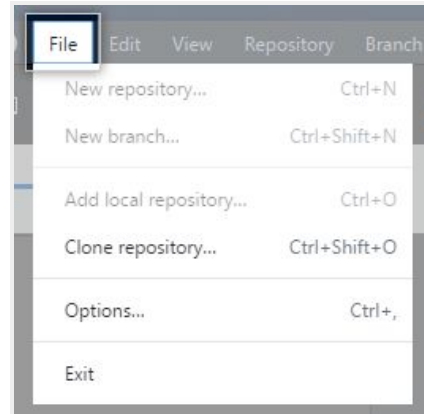
2.4.4 Configuring Git for GitHub Desktop

If you don't already have Git installed, you must configure it before using GitHub Desktop

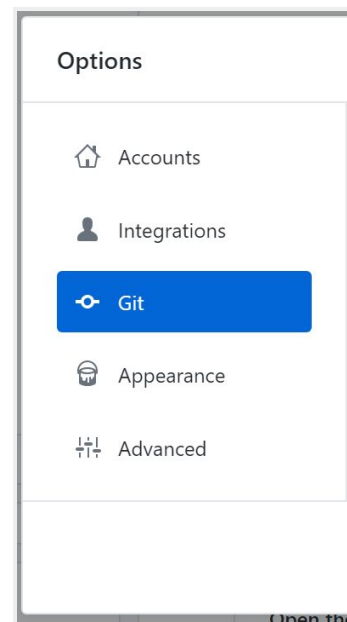
1. Sign in to GitHub or GitHub Enterprise.
2. In the upper-right corner of any page, click your profile photo, then click Settings.
3. In the left sidebar, click Emails.



4. Copy the email address you'd like to use in your local Git configuration.
5. Return to GitHub Desktop.
6. In the upper-left corner of the window, select the File menu.
7. Click Options.



8. In the Options window, select Git.



9. In the Name field, type the name you'd like to use for your Git configuration.

The screenshot shows a dialog box titled "Options" with a close button (X) in the top right corner. On the left side, there is a vertical sidebar with five menu items: "Accounts" (with a house icon), "Integrations" (with a person icon), "Git" (with a Git logo icon and highlighted in blue), "Appearance" (with a paint palette icon), and "Advanced" (with a gear icon). The main area of the dialog has two input fields. The "Name" field contains the text "Monalisa Octocat". The "Email" field is currently empty. At the bottom right of the dialog, there are two buttons: a blue "Save" button and a white "Cancel" button with a grey border. Below the dialog box, there is a faint, partially visible text: "(Open the repository page on GitHub in your browser)".

10. In the Email field, paste the email address you copied from your GitHub or GitHub Enterprise account settings.

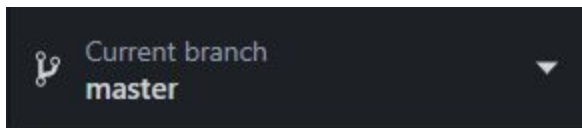
This screenshot is identical to the previous one, showing the "Options" dialog box. The "Name" field still contains "Monalisa Octocat". However, the "Email" field now contains the email address "monalisa@github.com". The "Save" and "Cancel" buttons remain at the bottom right. The same faint text "(Open the repository page on GitHub in your browser)" is visible below the dialog.

3. Project Setup

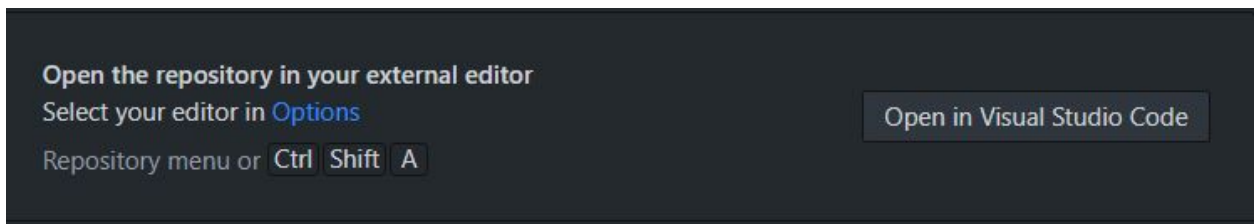
3.1 Loading the Project

To load the project, open the Github Desktop application.

Ensure that you are on the 'master' branch, located in the top center square of the application screen.



On the right hand side of the screen, there should be text saying 'Open the repository in your external editor. The button next to that text should read 'Open in Visual Studio Code'.



If it does not, then you need to click on the blue 'Options' text to the left of the button. Click 'Integrations' and select Visual Studio Code in the drop down list under 'External Editor'.

3.2 Importing Dependencies

4. Running the Program

4.1 Running on Localhost

In the editor of your choice you need to have Node.js installed. This should have been done in section 2.

Load up the project as shown in section 3.

Then open up the terminal, type 'node ./app.js' and press enter as seen below:



Open up your web browser of choice, then navigate to '<http://localhost:5000/>'.

4.2 Viewing the ClearDB database on MySQL Workbench

Open MySQL workbench

Under the Toolbar click the plus button:

MySQL Connections

Edit the form in the following ways:

Connection Name: Mercy Clinic Heroku

Hostname: us-cdbr-iron-east-04.cleardb.net

Username: ba042d12c1b782

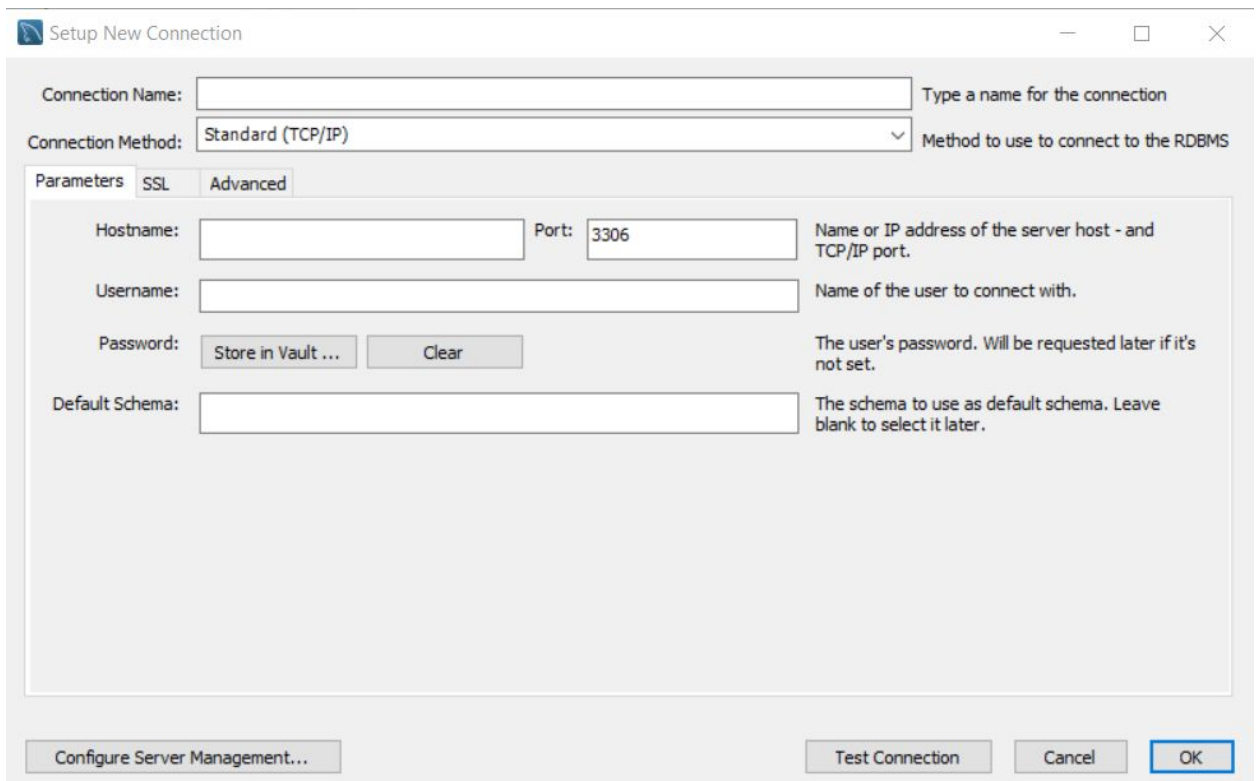
Default Schema: heroku_01f7a85e562f7f6

Then press 'ok'

Afterwards a new connection will appear on the MySQL home screen. Double click it.

It will prompt for you to enter a password, enter: 'add password here'.

Open up a new query to start editing and viewing the database.



Setup New Connection

Connection Name: Type a name for the connection

Connection Method: Standard (TCP/IP) Method to use to connect to the RDBMS

Parameters SSL Advanced

Hostname: Port: 3306 Name or IP address of the server host - and TCP/IP port.

Username: Name of the user to connect with.

Password: Store in Vault ... Clear The user's password. Will be requested later if it's not set.

Default Schema: The schema to use as default schema. Leave blank to select it later.

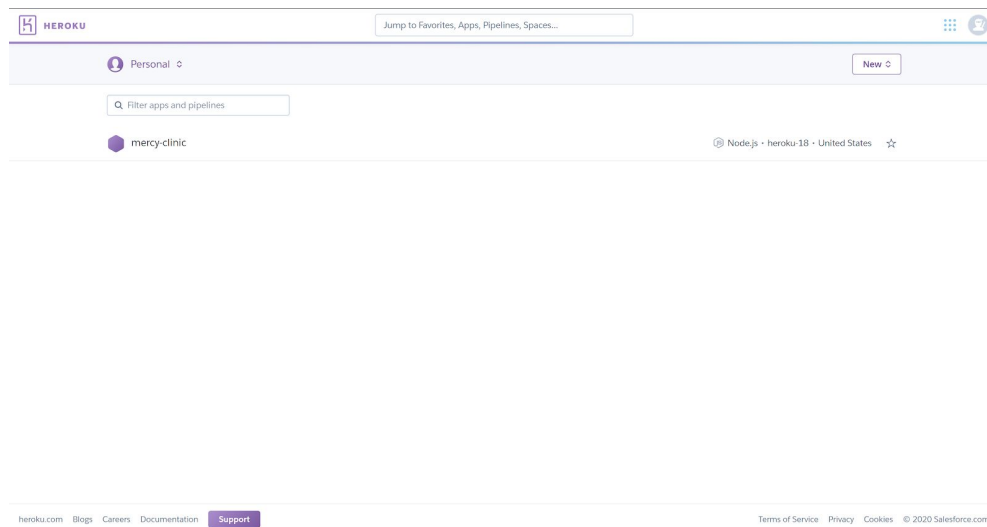
Configure Server Management... Test Connection Cancel OK

5. Deployment and Heroku Dyno Configuration

5.1 Setting up Heroku Dyno

Travel to <https://id.heroku.com/login> to access the Heroku account. (See Further Resources for information about obtaining access).

After logging in, you will arrive at the Heroku online dashboard:

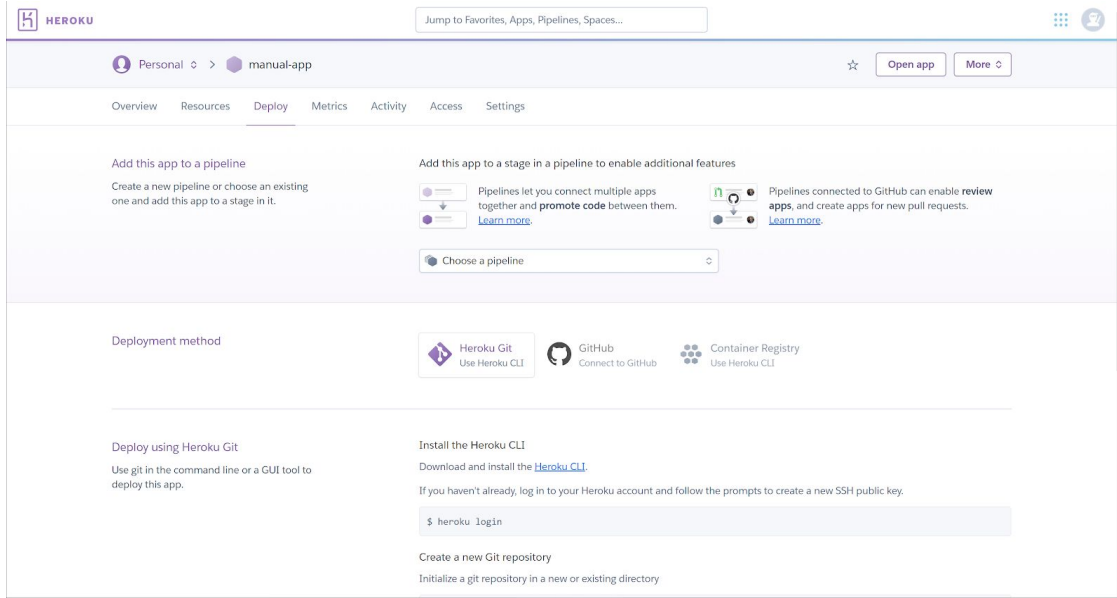


Any dynos that the Heroku account is currently maintaining will be listed in the dashboard, such as the mercy-clinic app.

To create a new dyno (platform that will host and run your application), click 'New' in the top right hand corner of the dashboard. Then click 'Create new app'.

On the next screen, fill in the name for your dyno (like mercy-clinic) and then select the region the app is being deployed from (where you are). Then click 'Create new app'.

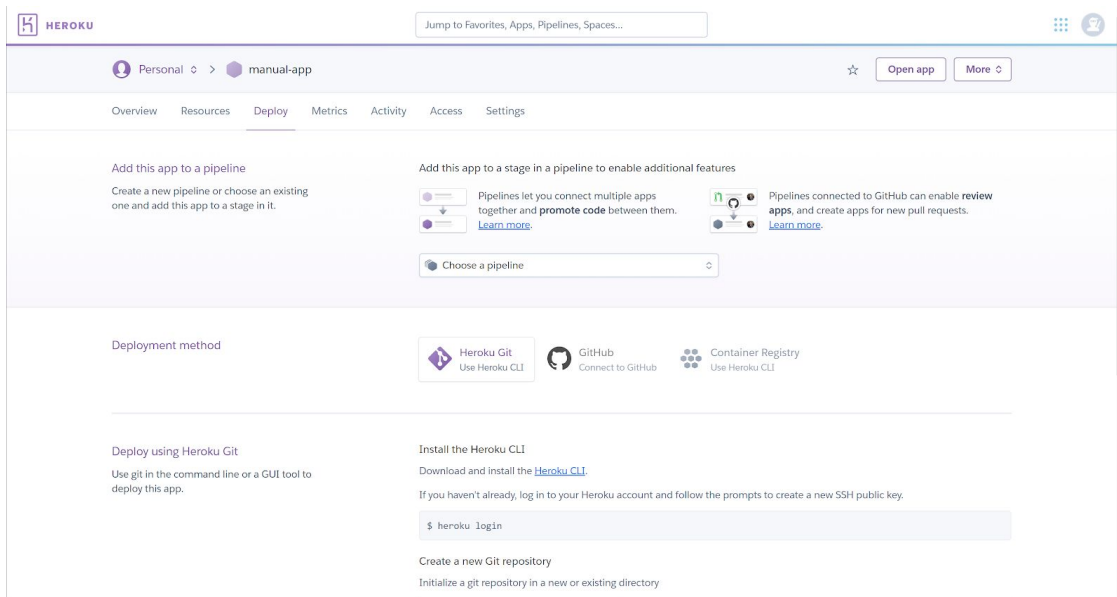
After this, you will arrive at the dashboard for the application:



At this point, you have completed the setup of the Heroku Dyno. To connect your application code and deploy the application, see section 5.2.

5.2 Deploying Application on Heroku

To recreate our project, we chose to deploy the application by connecting the dyno to our Github repository. Start from the dyno dashboard (see section 5.1):



Click on the 'Github - Connect to Github' button listed as one of three deployment methods (the symbol is a white cat in a black circle). You will need to authenticate with Github. You only need to do this one time per account.

Connect to GitHub

Connect this app to GitHub to enable code diffs and deploys.

Search for a repository to connect to

Missing a GitHub organization? [Ensure Heroku Dashboard has team access.](#)

Once you have authenticated your Github account, content will appear below that looks like the above image. Type in the name of your github repo (mercy-clinic) and click 'Search'.

Once it finds the correct repo, click Connect. Then you can choose your method of deployment, automatic or manual. Our project chose to set up automatic deployment so that the site can run continuously and any crashes would result in an automatic redeployment. To do this, simply select 'Enable Automatic Deploys', ensuring that 'master' is the branch selected in the drop down.

Automatic deploys

Enables a chosen branch to be automatically deployed to this app.

Enable automatic deploys from GitHub

Every push to the branch you specify here will deploy a new version of this app. **Deploys happen automatically:** be sure that this branch is always in a deployable state and any tests have passed before you push. [Learn more](#).

Choose a branch to deploy

Wait for CI to pass before deploy

Only enable this option if you have a Continuous Integration service configured on your repo.

Enable Automatic Deploys

Manual deploy

Deploy the current state of a branch to this app.

Deploy a GitHub branch

This will deploy the current state of the branch you specify below. [Learn more](#).

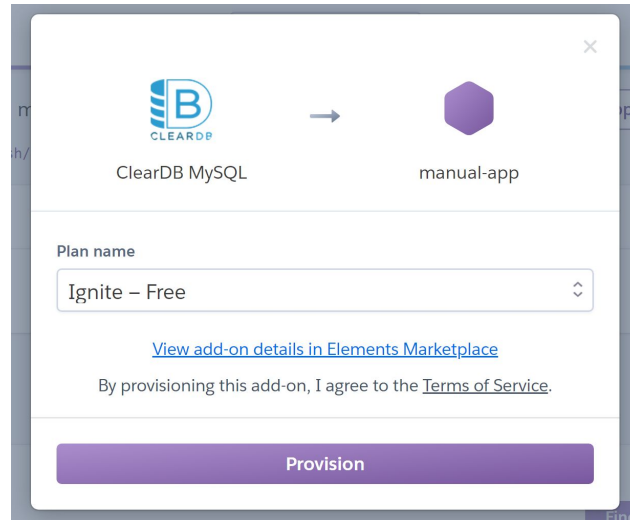
Choose a branch to deploy

Deploy Branch

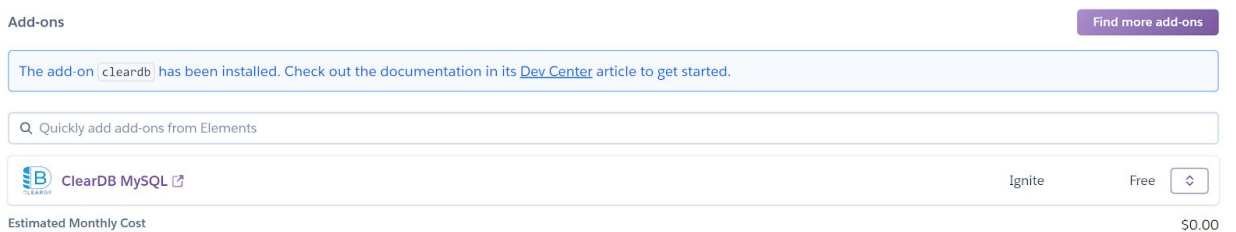
Manual deployments are useful during initial testing for one group of developers to test the system and identify weaknesses before pushing changes and continuing the test.

5.3 Connecting ClearDB to dyno

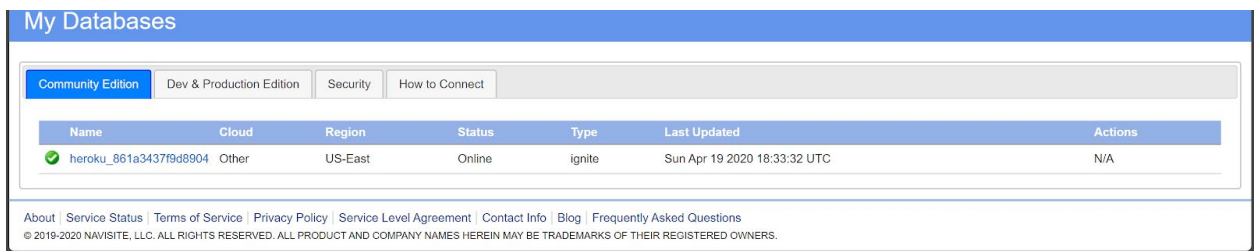
For this project, we use the ClearDB cloud database to hold the information necessary to maintain the mercy clinic volunteer system. This is because the ClearDB integrates seamlessly with the Heroku Dyno and uses MySQL, which is a familiar language to the developing team.



To connect the ClearDB to the dyno, navigate to the 'Resources' tab on the dashboard. Under 'Add-ons', type 'ClearDB' into the search box. A popup window will appear - click 'Provision'.



Clicking on 'ClearDB MYSQL' will take you to the database dashboard:

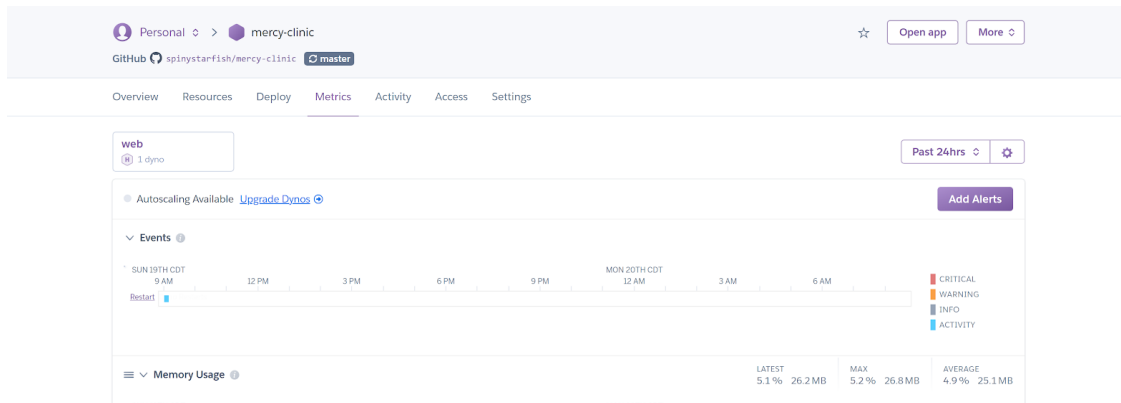


Congratulations, you have set up the ClearDB database with your Heroku dyno. To learn more about how to use ClearDB, check out the documentation in the Dev Center like the popup suggested.

5.4 Viewing Heroku Dyno Metrics

Any paid Heroku dyno comes with certain advantages. One of these advantages is a dash of site metrics that allows the developer to monitor certain resources and status of the application as it runs on the dyno.

To access the metrics page, please click on 'Metrics' in the menu tab of the dyno dashboard (see section 5.1).



This page can show you multiple statistics that are helpful in determining whether the site is running properly, handling the traffic it is receiving, and handling the amount of data it needs to store.

For more details on the metrics available for dyno's and what they mean, visit <https://devcenter.heroku.com/articles/metrics>.

6. Further Resources

6.1 SSL Certificate Monitoring

A benefit of a paid Heroku account is the automatic SSL certificate authentication feature. Heroku will set up and maintain a SSL certificate for your dyno. In addition, https is maintained, either by using the domain given by heroku or by using your own paid domain.

6.2 Super Admin Account

For the purpose of this project, a super user was created to give the developers usage in the system that a regular admin might not need. This super admin has the ability to edit and delete other admin users. Most importantly, this user cannot be deleted. This is to prevent lockout within the system, as a new admin can only be created by an admin. For this reason, it is very important that the **DATABASE SHOULD NEVER BE COMPLETELY WIPED**. The database should also not be directly altered such that the admin user is deleted.

In the event that such a thing does occur, a new super admin can be created with the same credentials manually using an SQL query and the site will respond to this user. Please use this information carefully: **creating a duplicate super user will most likely crash the site.**

Super user

id: 22 username: admin password: *create your own secure password*