



Title: User Manual

Date: April 28th, 2022

Olympic Developers

Peter Galvan, Camden Hortline,

Jason Gaglione, and Collin Rampata

Sponsor: Sean H. Anthony

Mentor: Melissa D. Rose

Overview: In this deliverable, the team clearly lays out the User Manual for the application. This will give information on Installation, Maintenance, and Trouble-shooting for the software.

1. Introduction	2
2. Installation	3
2.1 Installation of the Application	3
Pulling the Code From the Repository	3
Downloading NodeJS	4
Setting up the Frontend	4
Setting up the Backend	5
Setting up AWS Amplify	5
Running the Application Locally	6
Running the Application on the server	9
2.2 Installation of the AWS CLI for Admin User	11
Installing AWS CLI	11
Setting up User Accounts	12
2.3 Accessing the Database	12
Installing MySQL Workbench	13
Entering the Database	13
3. Maintenance	14
3.1 Database Maintenance	14
Complete Reset	14
After Deployment	15
4. Troubleshooting	15
4.1 Frontend Troubleshooting	15
4.2 Backend Troubleshooting	16
5. Conclusion	16

1. Introduction

This document will be taking you through how the application will be able to be installed, troubleshot, and maintained after the team has moved on from this product. This will help with the onboarding process for the application if the client decided to do any future work or hire a team to do so. There are other ways to install the application, perform maintenance and even troubleshoot the application. However, what is listed below is what the team has done during the capstone project. If an addition is made to the application, this document will need to be updated by the respective person who made an addition to the software.

2. Installation

This section will go over how to install the application and the AWS CLI for future work.

2.1 Installation of the Application

This section is about installing the application to the user's computer and having it run properly for a Windows machine. Please follow the steps in order to install the application.

Pulling the Code From the Repository

The code for the HYPO2 Management Web Portal is currently located in this [repository](#). To retrieve this code and have it on your device, the team recommends installing [GitHub desktop](#). There are other alternatives, however, the team found this way much easier.

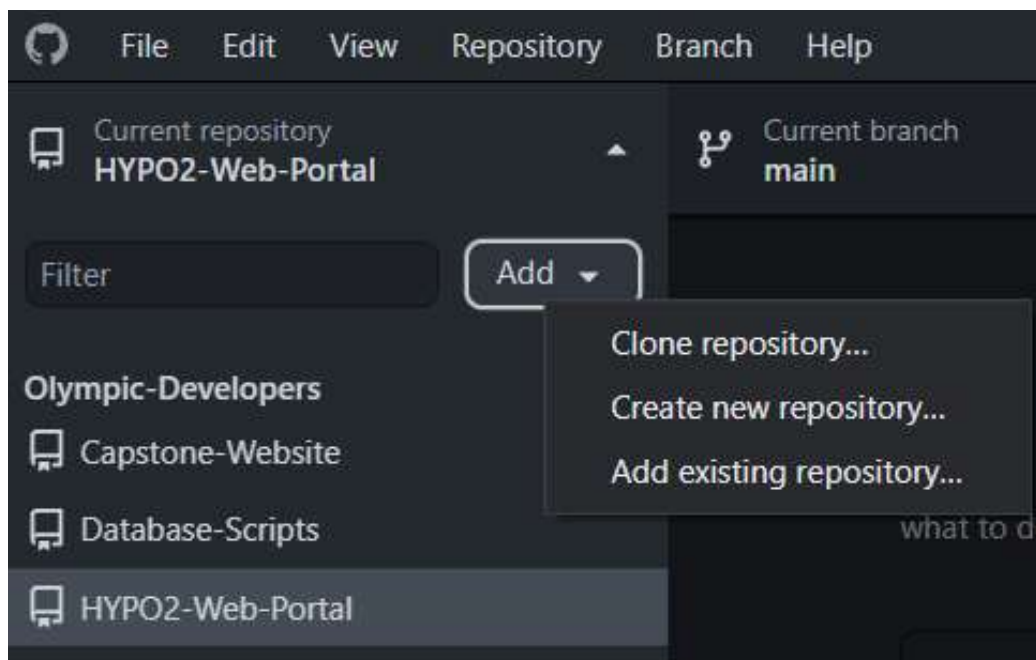


Figure 1: GitHub Desktop

When installing and logging into GitHub Desktop with a Github account a similar interface will be presented (Figure 1). To clone the repository, click the add button and then click Clone repository. Inside of Clone repository, go to the URL tab and code and paste the link to the repository above. Click on Clone. After downloading is done, the code will be available on the computer.

Downloading NodeJS

A download for [NodeJS](#) will be needed. Go to the link and download the software. This will be used to make sure the frontend and backend can properly work.

Setting up the Frontend

The team has used visual studio code. This can be downloaded as an IDE [here](#). From this part forward, the document will be assuming that visual studio code will be used. The next step is opening up the project folder HYPO2-WEB-PORTAL which was installed from the repository above.

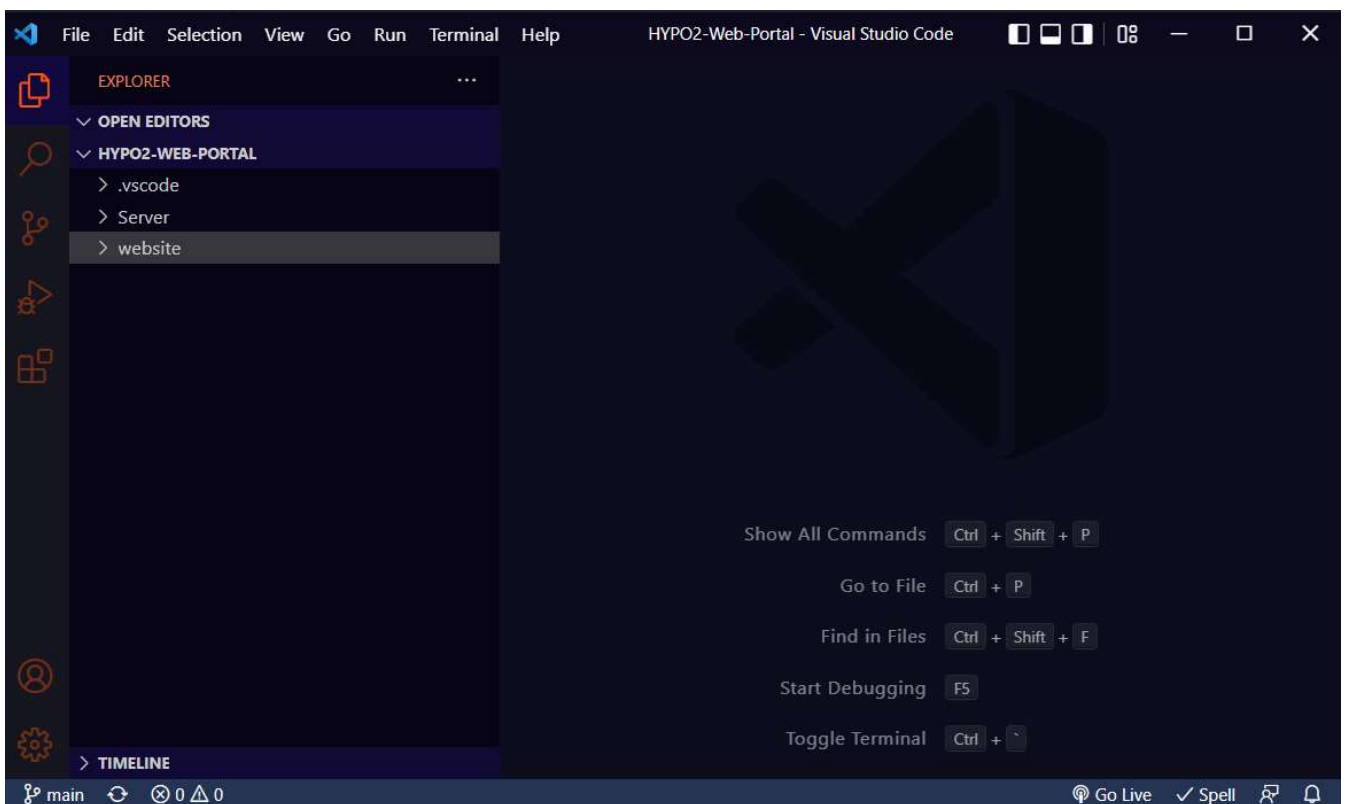


Figure 2: Image of starting interface of Visual Studio Code

Once open, a similar interface is seen above (Figure 2). Next, use the shortcut (ctrl + shift + `) to open the terminal window of the application.



```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
PS C:\Users\Peter Galvan\Documents\Github\HYPO2-Web-Portal> cd .\website\
PS C:\Users\Peter Galvan\Documents\Github\HYPO2-Web-Portal\website> npm install

```

Figure 3: Terminal Window in Visual Studio Code

Inside the terminal window, insert the following two commands displayed above (Figure 3). After the download is done, inside the website folder there should be a provided folder called `node_modules`. The frontend of the application should be deployed.

Setting up the Backend

The backend for this application will already be set up if the above steps were followed correctly for pulling the code from the repository and downloading NodeJS.

Setting up AWS Amplify

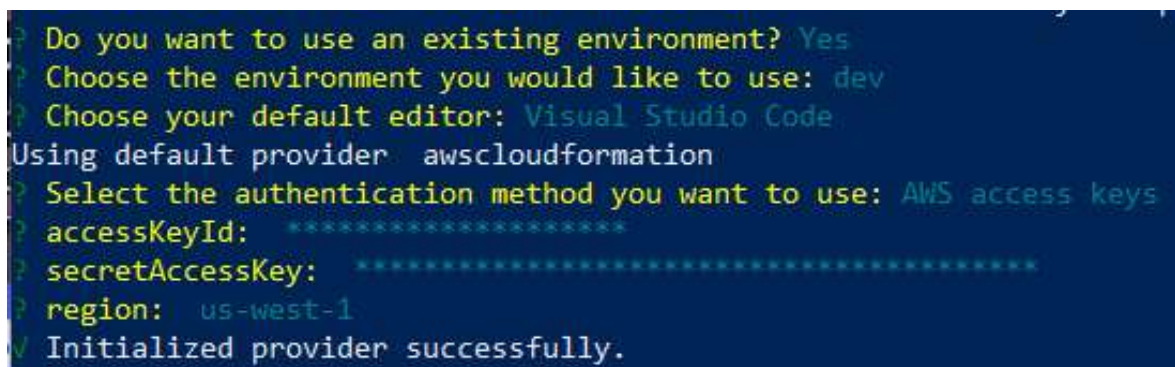
All developers of the application used Windows. So, this will assume Windows is being used, but MAC can be utilized as well. However, these steps are for Windows users only. You need to open up PowerShell and make sure to run this application as an administrator. Run the following command below (Figure 4).



```
npm install -g @aws-amplify/cli
```

Figure 4: Install command for AWS Amplify CLI

Once this is done installing move to the directory `...\HYPO2-Web-Portal\website`. A command `amplify init` will need to be run as soon as this point is reached.



```

? Do you want to use an existing environment? Yes
? Choose the environment you would like to use: dev
? Choose your default editor: Visual Studio Code
Using default provider  awscloudformation
? Select the authentication method you want to use: AWS access keys
? accessKeyId: *****
? secretAccessKey: *****
? region: us-west-1
✓ Initialized provider successfully.

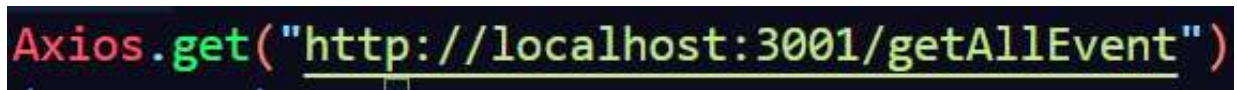
```

Figure 5: Amplify init setup

Follow the steps above(Figure 5). The accessKeyID is AKIAXDSOSM5DCCTVUMU6 and the secretAccessKey is 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. Once this is successfully installed make sure that the directory you currently are in now has a file aws-exports.js if it does AWS Amplify was properly installed.

Running the Application Locally

To run the application locally, check if the Axios.get, Axios.post, and Axios.delete are all directed to the correct location.



```
Axios.get("http://localhost:3001/getAllEvent")
```

Figure 6: Axios.get for local use

The above is an example (Figure 6) from the file AdminProfile.js for an Axios.get function which is located in the folder ...\\HYPO2-Web-Portal\\website\\src\\Pages\\Profile Pages. The “/getAllEvent” changes what the function is calling. For example, in the same directory but in the file ClientProfile is “/UserCamps”. Also, make sure that the app.post, app.get, app.delete are directed to the correct location.



```
app.get("/getAllEvent",
```

Figure 7: app.get for local use

The above is an example (Figure 7) from the file index.js for an app.get function which is located in the folder ...\\HYPO2-Web-Portal\\server. The “/getAllEvent” changes for what the function is calling. For the function underneath “/getAllEvent” it will use “/getUserEvent” instead. Once all the Axios and app functions are changed for local use, open the terminal and go to the directory ...\\HYPO2-Web-Portal\\website. Type ‘npm start,’ this will open the browser once loaded (Figure 8) will be seen, which is below.

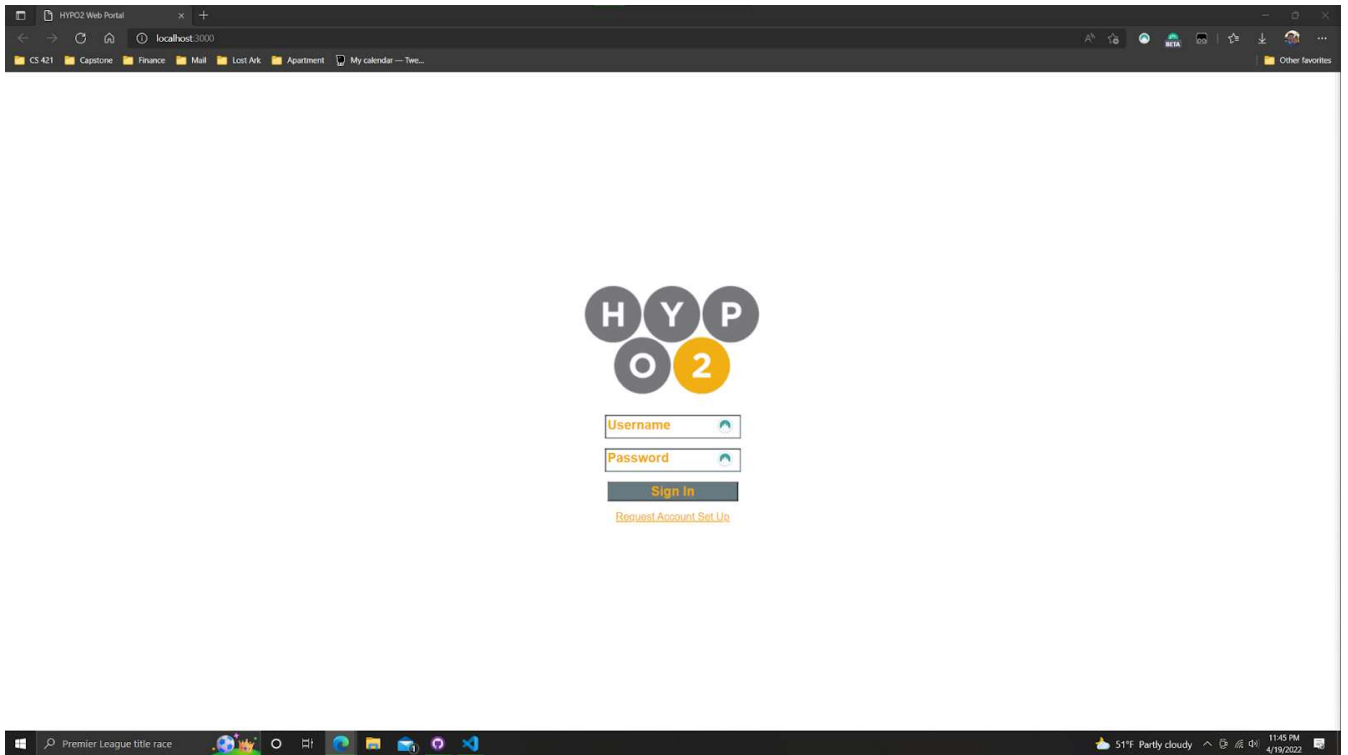


Figure 8: HYPO2 Web Portal Login Screen

From here, any created account in the system can be logged in. Below (Figure 9), is a list of all the test users the team made. It can be logged in to check if the application is running correctly.

```
Staff Testing Massage
Username: testingStaffUserMassage
Password: Testing#1

Admin Testing
Username: testingAdminUser
Password: Testing#2

Client Testing
Username: testingClientUser
Password: Testing#3

Client Testing 2
Username: testingClientUser2
Password: Testing#4

Staff Testing Sport Psychology / Mental
Performance
Username: testingStaffUserMental
Password: Testing#5

Staff Testing Blood Testing & Physiological Testing
Username: testingStaffUserBloodPhy
Password: Testing#6

Staff Physiotherapy/Chiropractic Rehab/Prehab &
Strength & Conditioning Coaching testing
Username: testingStaffUserPreCoach
Password: Testing#7

Staff Physiotherapy/Chiropractic Rehab/Prehab &
Strength & Conditioning Coaching & Sport
Nutrition testing
Username: testingStaffUserPreCoachNutrition
Password: Testing#8
```

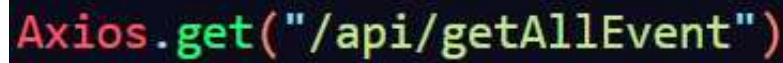
Figure 9: All testing User Accounts

Once reaching the above (Figure 8), the frontend should be working correctly. The AWS Amplify should be set up correctly once logged in. Finally, if camps are populated in the

database and the respective users can see the information, the backend is working properly as well. The application now is running locally on the machine.

Running the Application on the server

Currently, the code is running on the server [here](#). This is only needed if there is a want or need to make any changes to what is currently running. Also, it is always recommended to check that the code runs locally before moving it to the server. To run the application on the server, check if the Axios.get, Axios.post, and Axios.delete are all directed to the correct location.



```
Axios.get("/api/getAllEvent")
```

Figure 10: Axios.get for server use

The above is an example (Figure 10) from the file AdminProfile.js for an Axios.get function which is located in the folder ...\\HYPO2-Web-Portal\\website\\src\\Pages\\Profile Pages. The “/getAllEvent” changes on what the function is being called. For example, in the same directory, but in the file, ClientProfile is “/UserCamps”. Also, make sure that the app.post, app.get, app.delete are directed to the correct location.



```
app.get("/api/getAllEvent",
```

Figure 11: app.get for local use

The above is an example (Figure 7) from the file index.js for an app.get function which is located in the folder ...\\HYPO2-Web-Portal\\server. The “/getAllEvent” changes for what the function is calling for. For the function underneath, “/getAllEvent” it will use “/getUserEvent” instead. Once all the Axios and app functions are changed for server use, open the terminal and go to the directory ...\\HYPO2-Web-Portal\\website type npm run build. This will create a folder called “build” in the directory in the same location as the current directory. Once all the files are downloaded to the folder “build,” the files will need to be transferred to the server. This team used FileZilla which can be downloaded [here](#). From this part forward, the document will be assuming FileZilla is being used. On installation open, the application, click on the file in the top right corner then click on site manager click on the new site (Figure 12) will be visible on the right.

General Advanced Transfer Settings Charset

Protocol: FTP - File Transfer Protocol

Host: Port:

Encryption: Use explicit FTP over TLS if available

Logon Type: Normal

User:

Password:

Background color: None

Comments:

Connect OK Cancel

Figure 12: FileZilla Site Manger Interface

Change the Protocol to SFTP - SSH File Transfer Protocol, for the Host enter “ec2-54-67-109-121.us-west-1.compute.amazonaws.com”, change Logon Type to Key file, and the User will be Ubuntu. The Key file will be where the .ppk key is placed. This was provided to the client in the USB he was given at the end of the capstone to get the key from him. Finally, click connect, which will then connect to the server. Before moving over any files to the server, the team highly encourages taking the HYPO2WebPortal Folder from the server and storing it safely somewhere on the computer. If the code causes any errors on the server, go to the folder HYPO2WebPortal the interface is shown below (Figure 13).

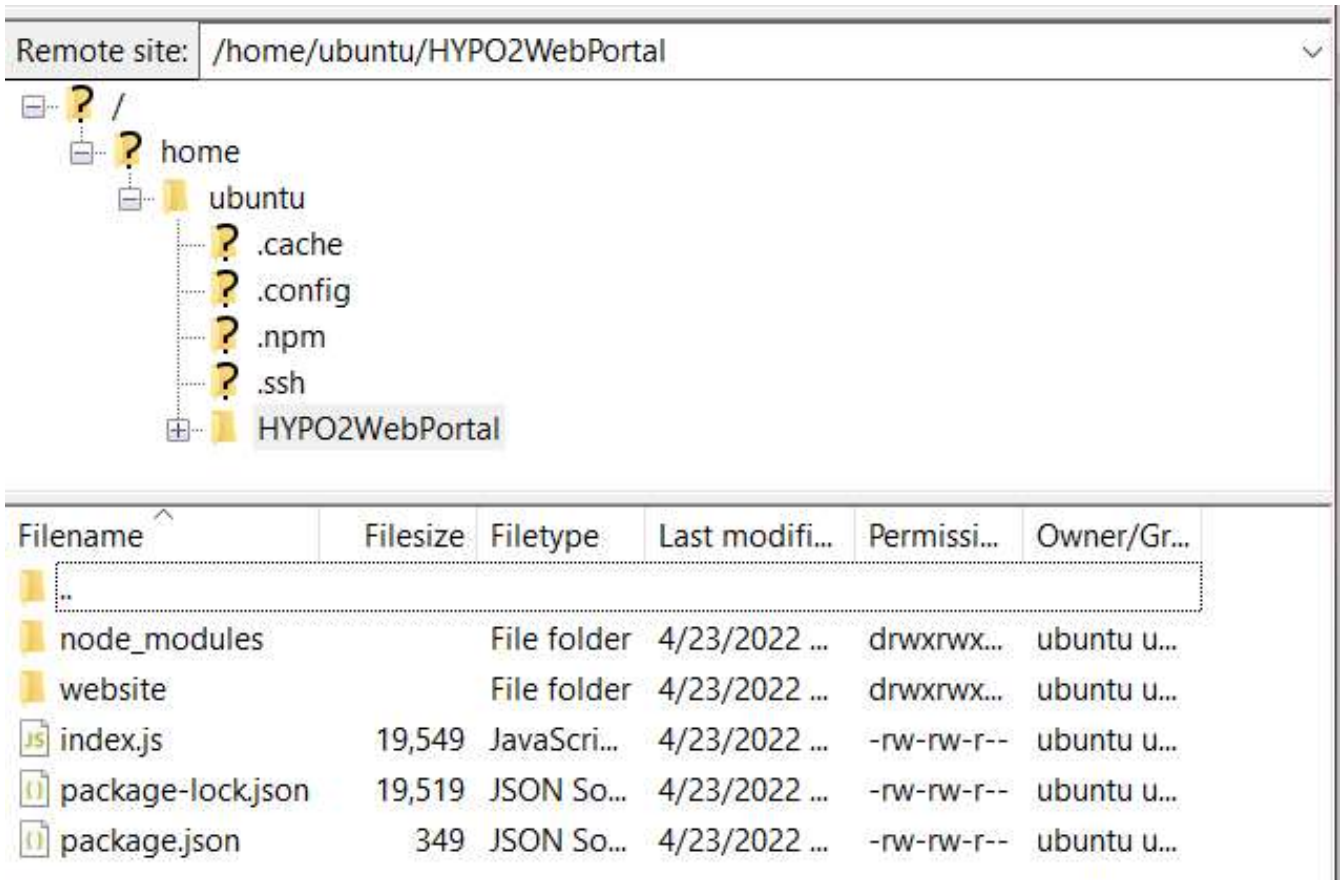


Figure 13: FileZilla on the Server inside HYPO2WebPortal

Here, two folders will be seen. If any changes were made to the Server file, open the Server file on a local machine on the right and transfer all the information into the server's current location from the right to the left (Figure 13). The node_modules folder here is just the modules used by the server. Next, go into where the "build" folder on the local machine on the right of what was made earlier and take all information from that folder and move it over to the website folder on the server. All the changes made should now be visible [here](#). If they are not, use (shift + r) to hard reset the website. If still not, there might be an error in the code moved over. If running, it can be tested similarly to how it was tested locally by using the test users in Figure 9.

2.2 Installation of the AWS CLI for Admin User

This section is about installing the AWS CLI to the user's computer and having it run properly.

Installing AWS CLI

Here AWS CLI will be installed by clicking [here](#). All developers of the application used Windows, so this will assume Windows is being used. It does have the capability to be done with MAC as well. However, these steps are for Windows users only. PowerShell needs to be opened up and run as an administrator. Now open Powershell and make sure to run it as an administrator.

Type in the command `AWSconfigure`. It will ask for an Access Key ID provided it with `AKIAXDSOSM5DNRMRKIG4` and then it will ask for a Secret Access Key provided it with `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`.for default region. Recommended for use is `us-west 1` and the default output format should include just clicking enter. The AWS CLI is now set up.

Setting up User Accounts

This will go over how to set up accounts using the AWS CLI. All developers of the application used Windows so this will assume that Windows is being used, but MAC can be used as well. However, these steps are for Windows users only. PowerShell needs to be opened up and run as an administrator. The team has created a document containing the setup for all types of users. There is an example of how to create a client user below (Figure 14). The document will be provided to the client on the USB given after the capstone.

Client User Setup

```
aws cognito-idp sign-up --client-id 2r2b250gso0fg67qivnnqehpo5 --username (Client Username)
--password (Password for Client User) --user-attributes Name="email",Value="(Email for Client
Users)" Name="custom:Classification",Value="Client"
```

Figure 14: Creating a Client User

Information will now be put in the () remove () before running the command in Powershell. The next step will be to confirm the user with the code below (Figure 15).

All User Setup

```
aws cognito-idp admin-confirm-sign-up --user-pool-id us-west-1_mENwHCQU4 --username
(Username of the user you are trying to confirm)
```

Figure 15: Confirming a user

When confirming a user, make sure the same Username as the Username created originally is inputted. Now creating an account is accessible. For more users, visit the document provided to the client.

2.3 Accessing the Database

This section is about accessing the database on the user's computer and having it run properly.

Installing MySQL Workbench

For interacting with the database, the team used MySQL Workbench. This is what the team recommends using. It can be accessed using other methods. However, from this point on the team will be assuming the use of MySQL Workbench. A download for [MySQL Workbench](#) will be needed. Go to the link and download the software.

Entering the Database

Once it has downloaded MySQL Workbench, open the program. There needs to be created a connection to the database. To do this, click on the plus icon next to MySQL Connections. This looks like Figure 16 below.



Figure 16: MySQL Connections

Once the plus has been clicked there will be given an interface asking you to create the connection below(Figure 17) that will show this interface.

 The image shows the 'Setup New Connection' dialog box in MySQL Workbench. The dialog has a title bar with a minus, maximize, and close button. It contains several fields and tabs:

- Connection Name:** A text input field with a placeholder 'Type a name for the connection'.
- Connection Method:** A dropdown menu set to 'Standard (TCP/IP)' with a placeholder 'Method to use to connect to the RDBMS'.
- Parameters tab:** This tab is selected, showing:
 - Hostname:** '127.0.0.1' and **Port:** '3306'. A note says 'Name or IP address of the server host - and TCP/IP port.'
 - Username:** 'root'. A note says 'Name of the user to connect with.'
 - Password:** A field with 'Store in Vault ...' and 'Clear' buttons. A note says 'The user's password. Will be requested later if it's not set.'
 - Default Schema:** An empty text field. A note says 'The schema to use as default schema. Leave blank to select it later.'
- Advanced tab:** Currently inactive.
- SSL tab:** Currently inactive.
- Buttons at the bottom:** 'Configure Server Management...', 'Test Connection', 'Cancel', and 'OK'.

Figure 17: MySQL Connection Setup Interface

For Connection Name put HYPO2 Web Portal, Connection Method is standard, Hostname is database-1.caabureps5is.us-west-1.rds.amazonaws.com, the port is on 3306 and the Username is Admin_hypo. If it ever asks for a password the password is sCuxOyRJE06qTWs8x5w however the team has never actually had to enter this password. The team thinks it is mostly used for connecting the database to NodeJS like in the directory ...\\HYPO2-Web-Portal\\Server file indes.js.

3. Maintenance

This section will go over how to maintain the application. The application does not need much maintenance and is useless if there is a want to reset the database for testing or remove a camp in the current deployment of the software.

3.1 Database Maintenance

Testing Camp information will be stored in the database for proper functionality. This data is not necessary and can fill the database with useless information. In order to properly maintain the database, certain scripts should be ran to keep the database clear.

Complete Reset

Before the application is deployed, the database should be completely reset after every testing period. This will save the database usage as well as remove any unnecessary information that might confuse users when the app is deployed.

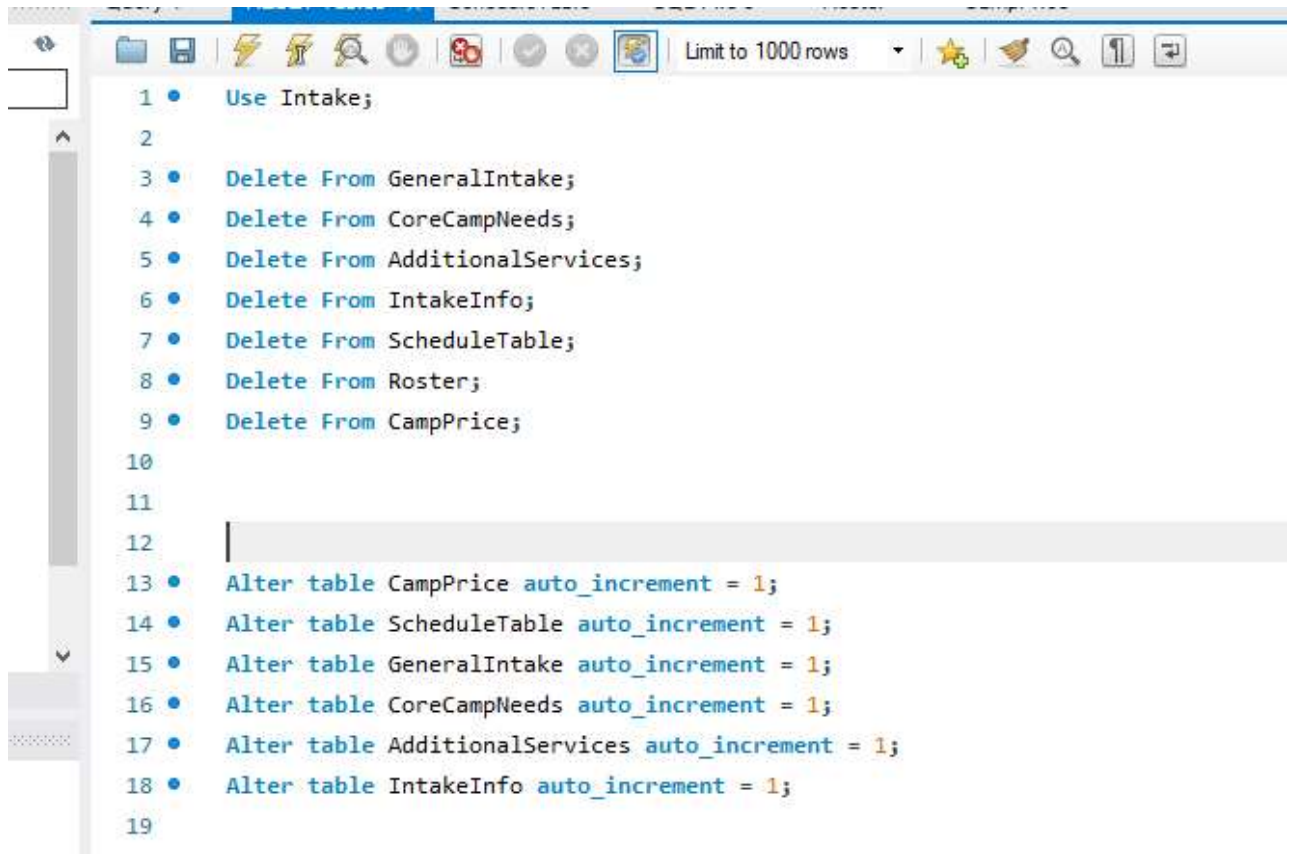


Figure 18: Database Reset Script

After Deployment

The application has the delete camp feature under the admin. If testing happens, a single test camp can be deleted within the application. To make sure the application is clear of test cases, the camp lists should be checked frequently and any camps that are from a test case should be deleted to delude confusion within the application.

4. Troubleshooting

This section goes into detail about troubleshooting the application. It will go over how to troubleshoot the frontend and backend issues.

4.1 Frontend Troubleshooting

The React frontend updates frequently new npm packages which are required during some updates. If method mentioned should be utilized if the code is attempting to be run, but will not start. In order to make sure the application has the most up-to-date protection and version of React modules, in the directory ...\\HYPO2-Web-Portal\\website run the command `npm install`. This will reinstall all node modules and add any missing modules needed for the application. The

team recommends making sure all node modules are downloaded and updated before building the application and pushing it to the server.

4.2 Backend Troubleshooting

There is not much troubleshooting for the backend if the application is not pulling any information from the database. Make sure the Axios function is pointed to the right place. For how to write this correctly, check Figure 9 for server deployment and check Figure 6 for local use.

5. Conclusion

This concludes the user manual. The team was very happy to work with you and wishes you many great years using this product and doing future work. While we are all moving on to professional careers, we would be happy to answer short questions in the coming months to help you get the product deployed and operating optimally in your organization. Thank you, Peter Galvan (Pjg94@nau.edu) Camden Hortline (cjh547@nau.edu), Collin Rampata (ccr243@nau.edu), and Jason Gaglione (jwg242@nau.edu).