

# 2023-CITS3200-Team26's Wave Dashboard

## Installation Guide:

The installation of dependencies is managed by a script called 'setup.sh'. It reads all the dependencies required by the project and installs them and performs the necessary database migrations to fit for use. To use, simply, open a terminal interface in the folder of the script, and run 'bash setup.sh' this will run all the necessary commands. If this does not work, here's how to install it the long way:

1. Install python.  
If not already done so, python is required to run the the website as the framework, Django, is based in python. To do so download the installer from the official website and follow the instructions.
2. Setup a virtual environment.  
Selecting the project folder as directory in a command line, execute the command 'python3 -m venv venv' to create the virtual environment, this means that all dependency installation will be done locally to prevent version conflict issues. To select the virtual environment, run 'source venv/bin/activate' to activate the virtual environment.
3. Install the dependencies.  
Python provides a shortened way of installing dependencies, install the dependencies with 'pip install -r requirements.txt' to install all the required dependencies for the project.  
In addition, we will install nginx as our web server. We use the command 'sudo yum install nginx'
4. Set up gunicorn and nginx  
The project comes with configuration file for nginx and gunicorn. We just need to move them to the appropriate locations.  
  
'sudo mv conf/backend /etc/nginx/sites-available'  
'sudo ln -s /etc/nginx/sites-available/backend /etc/nginx/sites-enabled'  
  
'sudo systemctl restart nginx' or 'sudo systemctl start nginx'  
'nginx -t'
5. Set up our Django application's database  
'python manage.py makemigrations'  
'python manage.py migrate'
6. Add HTTPS  
We'll be using certbot to add HTTPS.  
We will first install certbot using the tutorial on their [website](#).

```
'sudo ln -s /home/ec-2user/2023-CITS3200-Team26 /venv/bin /usr/bin/certbot'  
'sudo certbot --nginx --rsa-key-size 4096 --no-redirect'
```

## **Uninstallation**

To uninstall, run the script 'uninstall.sh' to uninstall all dependencies associated with the project. Then the project itself, along with the virtual environment you created, can be deleted from the system.

## **Running:**

Similarly, to the installation, a script called 'run.sh' has been provided to simplify the process, simply run 'bash run.sh' to execute this script and run the webserver. If this fails, this is an alternative.

1. Open the terminal in the backend folder.  
As put, you need to open the terminal so it selects the first backend folder a shown in the GUI. An alternative is to open the terminal and type 'cd backend' to select the correct folder.
2. Run 'gunicorn -c conf/gunicorn\_conf'

## **Setting up Credentials**

1. Visit the link '<domain-name>/account/register/'

Side Note: To ensure security, only a maximum of one user is ever allowed to register, in the event that you forget your developer credentials during testing, you can run `python manage.py flush` to reset the database.

## 2. Create Username and Password

ICRAR

### Registration

Username:

test1

Required. 150 characters or fewer. Letters, digits and @/./+/-/\_ only.

Password:

•••••

Password confirmation:

•••••

Enter the same password as before, for verification.


Register

3. Login & Click "Enable Two-Factor Authentication"
4. Select the Token Generator method
5. Scan QR code into your preferred authenticator application (i.e., Microsoft Authenticator, Google Authenticator)

ICRAR

## Enable Two-Factor Authentication

To start using a token generator, please use your smartphone to scan the QR code below. For example, use Google Authenticator.



Alternatively you can use the following secret to setup TOTP in your authenticator or password manager manually.

TOTP Secret: [LVT4TZJ7OKBANDDEKYFF2UXPWRAFYKUE](#)

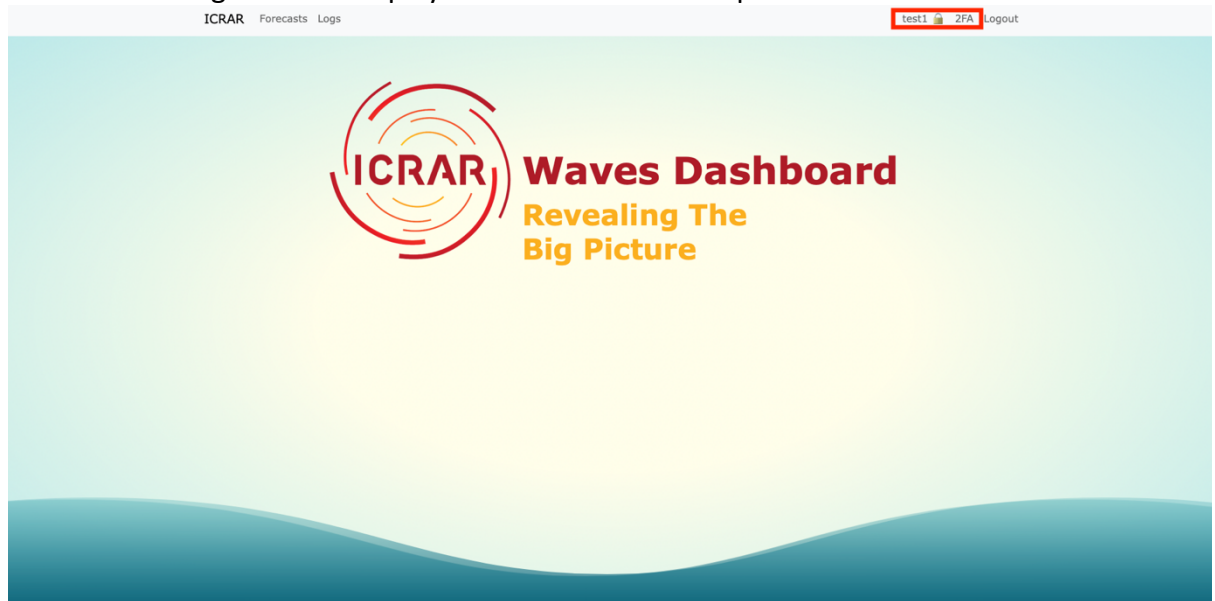
Then, enter the token generated by the app.

Token

BackNextCancel

6. Enter the OTP (updated every 30 seconds)

7. Padlock on navigation bar displays that 2FA has been implemented



## User Guide

The organization of the website is simple, with two pages of interest, **Forecasts** and **Logs**. Each page will also display file arrival times (more specifically, file modify times) below the link header bar for the data used by the ML model. This is added as a convenience so that it can be easily ascertained from any of the pages whether new files of interest have been received.

## Forecasts

The forecast page displays the significant wave height, peak period range, and wave direction from 3 data sources: actual buoy data, forecast from BOM, and the ML model.

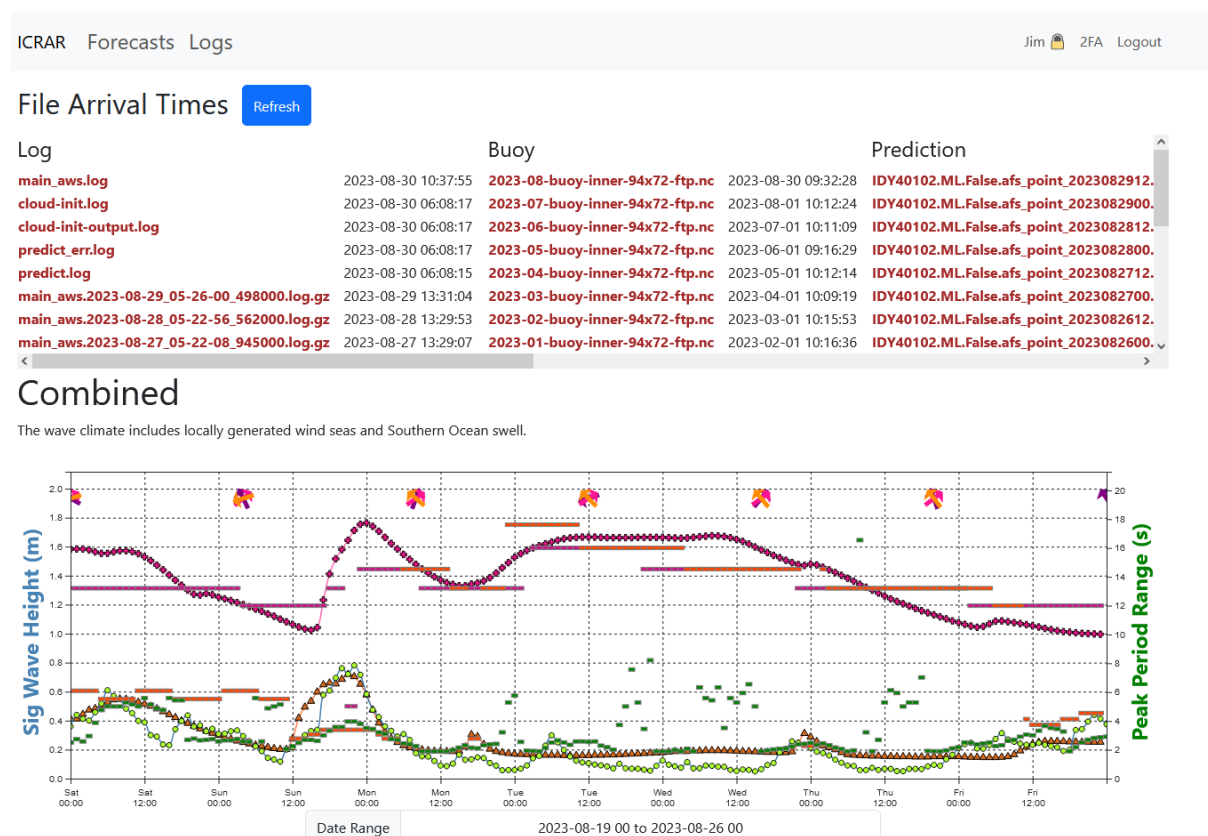


Figure 1. The Forecasts View

The usage of the graph is meant to be intuitive and interactive to use. To zoom in or out into any part of the graph, simply hover the cursor over the wanted region and scroll. Hovering over any data point also provides a tooltip, showing the exact date of the data point and the exact value of the data point.

By default, the graph will display the data from the current month. To select the time period to view data from, click on the wide bar that says "Select Date Range..", this will bring up a Date Selector menu, allowing you to choose the start and end date to view data from.

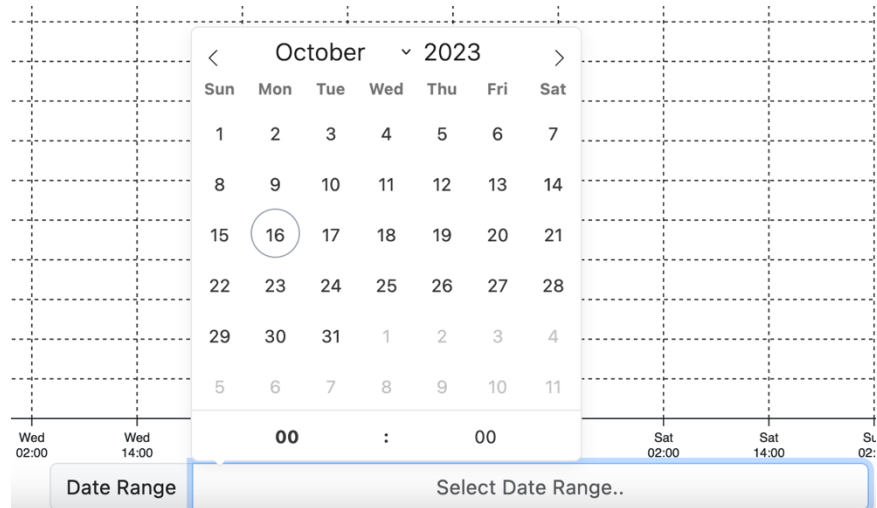


Figure 2. The Date Range View

## Logs

The logging page's main functionality is to display debug printed output. Two features of note are the File Selector drop down, found to the left, below the bolded header, and the Date Range selector, to select what period to display logs from.

File Arrival Times
Refresh

Log	Buoy	Prediction
main_aws.log	2023-08-30 10:37:55 2023-08-buoy-inner-94x72-ftp.nc	2023-08-30 09:32:28 IDY40102.ML.False.afs_point_2023082912.
cloud-init.log	2023-08-30 06:08:17 2023-07-buoy-inner-94x72-ftp.nc	2023-08-01 10:12:24 IDY40102.ML.False.afs_point_2023082900.
cloud-init-output.log	2023-08-30 06:08:17 2023-06-buoy-inner-94x72-ftp.nc	2023-07-01 10:11:09 IDY40102.ML.False.afs_point_2023082812.
predict_err.log	2023-08-30 06:08:17 2023-05-buoy-inner-94x72-ftp.nc	2023-06-01 09:16:29 IDY40102.ML.False.afs_point_2023082800.
predict.log	2023-08-30 06:08:15 2023-04-buoy-inner-94x72-ftp.nc	2023-05-01 10:12:14 IDY40102.ML.False.afs_point_2023082712.
main_aws.2023-08-29_05-26-00_498000.log.gz	2023-08-29 13:31:04 2023-03-buoy-inner-94x72-ftp.nc	2023-04-01 10:09:19 IDY40102.ML.False.afs_point_2023082700.
main_aws.2023-08-28_05-22-56_562000.log.gz	2023-08-28 13:29:53 2023-02-buoy-inner-94x72-ftp.nc	2023-03-01 10:15:53 IDY40102.ML.False.afs_point_2023082612.
main_aws.2023-08-27_05-22-08_945000.log.gz	2023-08-27 13:29:07 2023-01-buoy-inner-94x72-ftp.nc	2023-02-01 10:16:36 IDY40102.ML.False.afs_point_2023082600.

Logs
File management
Date Range
2023-08-27 00:00 to 2023-08-28 00:00
Search

☒ Ignore INFO logs
☐ Ignore tables

main\_aws.2023-08-28\_05-22-56\_562000.log.gz

Item	Value
file_locations_win64	{ 'data_directory': 'c:/kevin/work/ML-BOM/data_input/monthly_files', 'buoy_files': { 'inner': [ 'f:/bom/data/barrow/Wave_buoys/DWR_inner_Oct2020_Oct2021.nc', 'f:/bom/data/barrow/Wave_buoys/DWR_inner_Dec2019_May2020.nc', 'f:/bom/data/barrow/Wave_buoys/DWR_inner_v2.nc', 'f:/bom/data/barrow/Wave_buoys/Midi_buoy_Apr21_Mar22.nc' ], 'outer': [ 'f:/bom/data/barrow/Wave_buoys/DWR_outer_Sept2019_Dec2019.nc', 'f:/bom/data/barrow/Wave_buoys/DWR_outer_v2.nc' ] }, 'buoy_files/inner': [ 'f:/bom/data/barrow/Wave_buoys/DWR_inner_Oct2020_Oct2021.nc', 'f:/bom/data/barrow/Wave_buoys/DWR_inner_Dec2019_May2020.nc', 'f:/bom/data/barrow/Wave_buoys/DWR_inner_v2.nc', 'f:/bom/data/barrow/Wave_buoys/Midi_buoy_Apr21_Mar22.nc' ], 'buoy_files/outer': [ 'f:/bom/data/barrow/Wave_buoys/DWR_outer_Sept2019_Dec2019.nc', 'f:/bom/data/barrow/Wave_buoys/DWR_outer_v2.nc' ] }

Figure 3. The Logs View

There's an additional two option box below the selectors, which can filter out INFO outputs, leaving only the more pertinent outputs, and an additional box to ignore tables. Since removing the INFO logs will still display ASCII tables like the figure above.