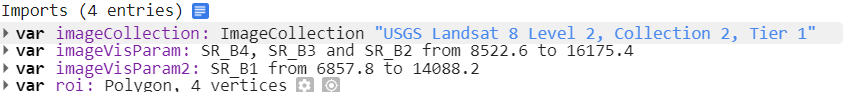
**Design Project – codes**

**MNDWI –**

[**https://code.earthengine.google.com/?scriptPath=users%2Fanisha232gf010%2Fpracticeweek2%3Afinal%20mndwi**](https://code.earthengine.google.com/?scriptPath=users%2Fanisha232gf010%2Fpracticeweek2%3Afinal%20mndwi)



var image = imageCollection

.filterDate("2022-11-11","2022-11-17")

.filterBounds(roi)

.mosaic()

.clip(roi);

Map.centerObject(roi, 11);

Map.addLayer(image,imageVisParam,'image');

//Calculate mndwi

var mndwi = image.normalizedDifference(['SR\_B3','SR\_B6']);

print(mndwi);

Map.addLayer(image,imageVisParam2,'mndwi');

//binary mask mndwi

var mndwi\_binary = mndwi.expression(

'b(0) > 0 ? 1 : 0',

{b: mndwi});

Map.addLayer(mndwi\_binary,{},'mndwi\_binary');

// Threshold for mndwi binary mask

var threshold = 0.05; // Adjust the threshold as needed

// Final mndwi with 0.05 threshold

var mndwifinal = mndwi\_binary.gt(threshold);

Map.addLayer(mndwifinal,{},'mndwifinal');

//Export to drive

Export.image.toDrive({

image: mndwifinal.toFloat(),

description: '2022-11-14',

scale: 30,

region: roi,

folder: 'test',

maxPixels: 1e13

});

**AWEI Code –**

[**https://code.earthengine.google.com/?scriptPath=users%2Fanisha232gf010%2Fpracticeweek2%3AAWEI**](https://code.earthengine.google.com/?scriptPath=users%2Fanisha232gf010%2Fpracticeweek2%3AAWEI)

**import geometry**

// Define parameters

var collection = ee.ImageCollection('LANDSAT/LC08/C02/T1\_L2')

.filterDate("2022-01-27","2022-02-01")

.filter(ee.Filter.bounds(geometry));

Map.centerObject(geometry, 11);

// Apply cloud mask

var maskL8sr = function(image) {

// Apply the scaling factors to the appropriate bands.

var opticalBands = image.select('SR\_B.')

.multiply(0.0000275).add(-0.2);

// Replace the original bands with the scaled ones

// and apply the masks.

return image.addBands(opticalBands, null, true)

};

var collection = collection.map(maskL8sr);

// Select and Rename Bands

var collection = collection.select(

['SR\_B2', 'SR\_B3', 'SR\_B4', 'SR\_B5', 'SR\_B6', 'SR\_B7'],

['blue', 'green', 'red', 'nir', 'swir1', 'swir2']

);

// Create a median composite and clip

var composite = collection.median().clip(geometry);

var rgbVis = {

bands: ['red', 'green', 'blue'],

min: 0.0,

max: 0.3

};

Map.addLayer(composite, rgbVis, 'Composite Image');

// Code to compute AWEI and detect water

// using Otsu thresholding

var detectWater = function(image) {

var awei = image.expression(

'4 \* (GREEN - SWIR1) - (0.25 \* NIR + 2.75 \* SWIR2)', {

'GREEN': image.select('green'),

'NIR': image.select('nir'),

'SWIR1': image.select('swir1'),

'SWIR2': image.select('swir2'),

}).rename('awei');

// Otsu Thresholding

var thresholding = require(

'users/gena/packages:thresholding');

var scale = 100;

var bounds = geometry;

var cannyThreshold = 0.7;

var cannySigma = 1;

var minValue = -0.2;

var th = thresholding.computeThresholdUsingOtsu(

awei, scale, bounds,

cannyThreshold, cannySigma, minValue);

// Create a Land-Water Image using Otsu Threshold

// You can replace th with a manual threshold if

// Otsu results are not satisfactory

var water = awei.gt(th).rename('water');

return water;

};

var water = detectWater(composite);

var waterVis = {min:0, max:1, palette: ['white', 'blue']};

Map.addLayer(water, waterVis, 'All Water');

//Export to drive

Export.image.toDrive({

image: water.toFloat(),

description: '2022-01-30',

scale: 30,

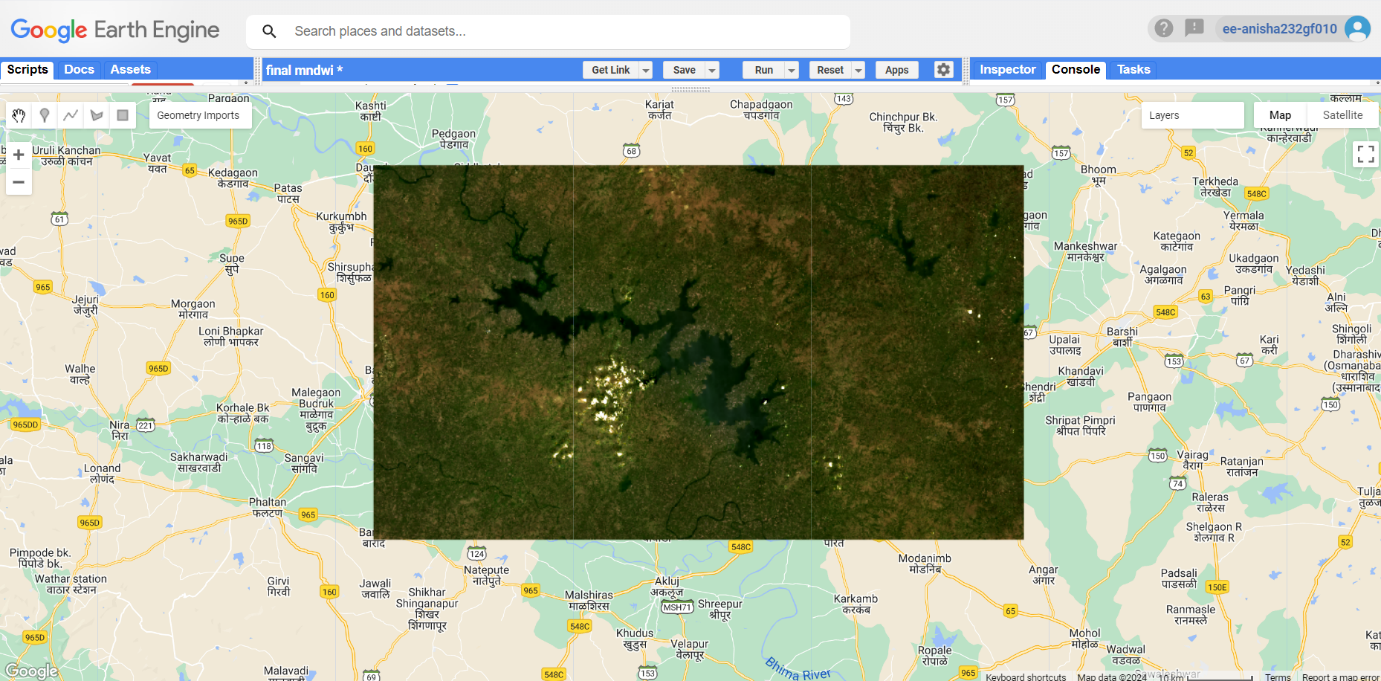
region: geometry,

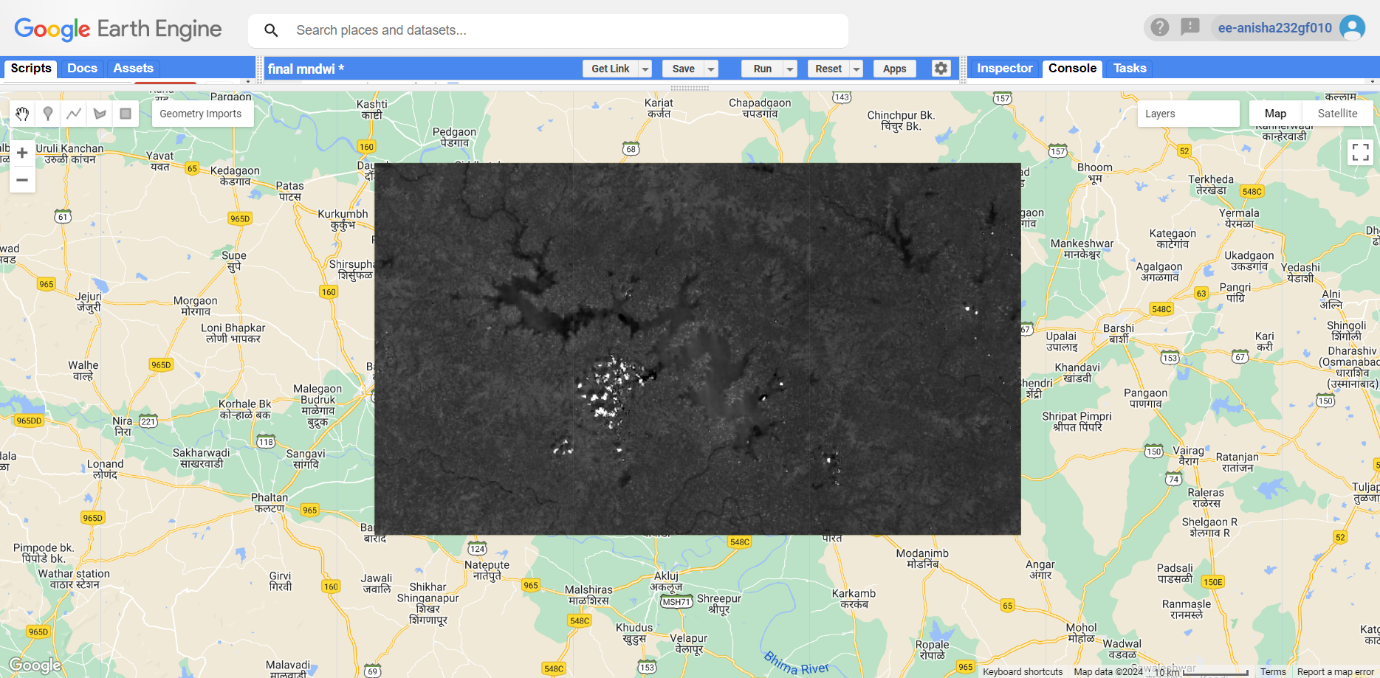
folder: 'test',

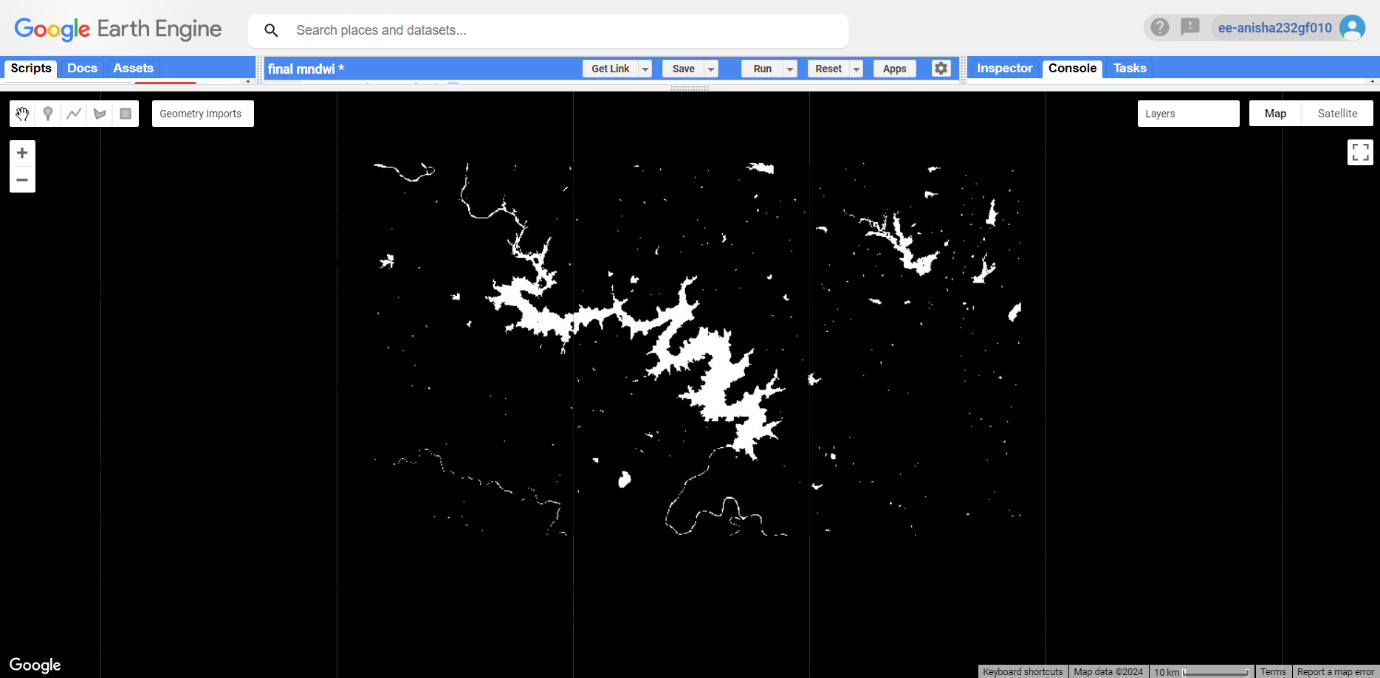
maxPixels: 1e13

});

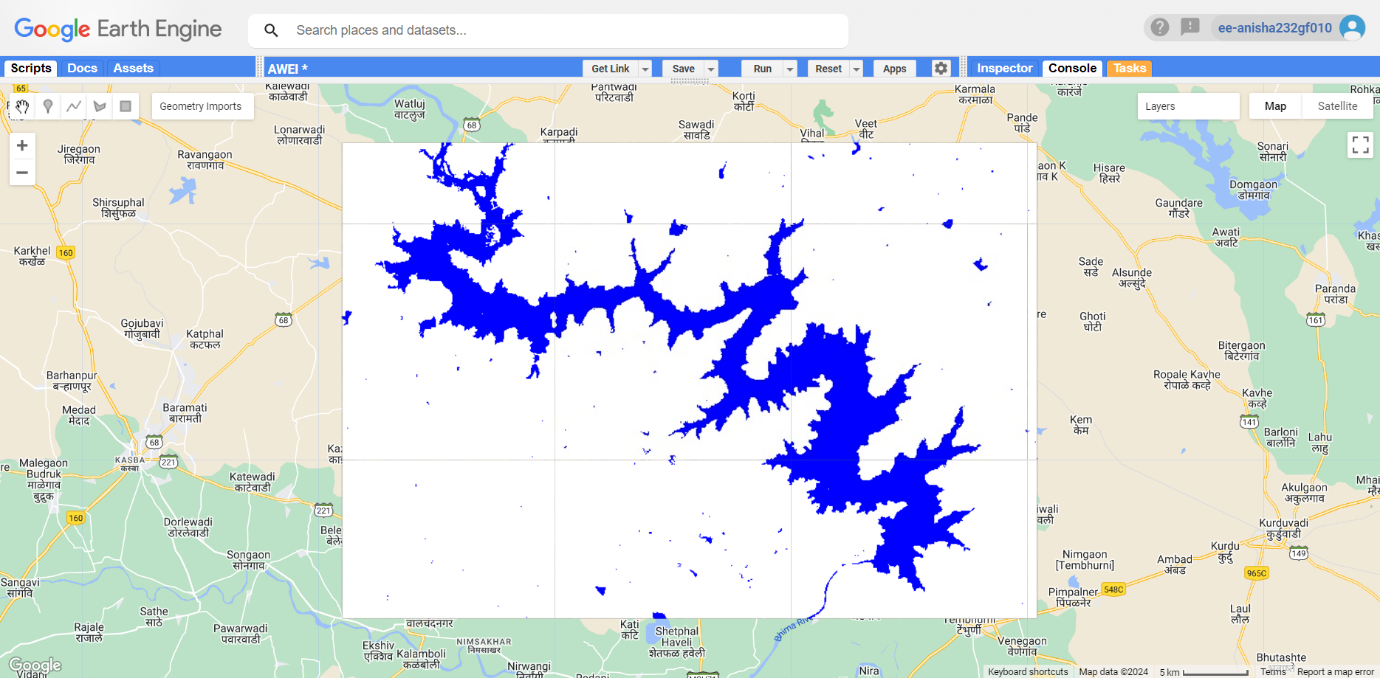
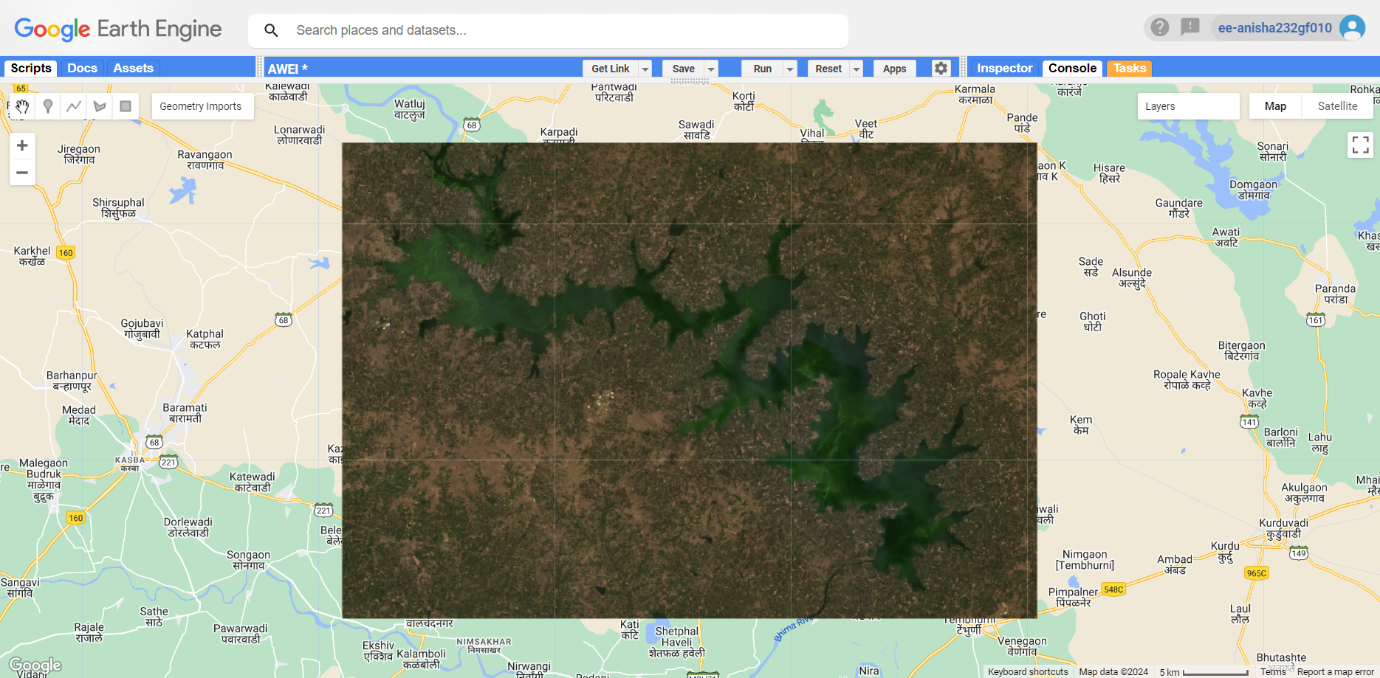
**MNDWI Outputs**

****

****

****

**AWEI Outputs**

****