// Set block

var block\_name = "Rajnagar";

var block\_name\_lower = block\_name.toLowerCase();

var block = ee.FeatureCollection("projects/ee-aatif/assets/MWS/" + block\_name + "\_MWS");

// LULC loading

var lulc\_imgs = [];

for (var y = 2017; y <= 2022; y++) {

var img = ee.Image('projects/ee-corestackdev/assets/datasets/LULC\_v3\_river\_basin/pan\_india\_lulc\_v3\_2017\_2018');

// var img = ee.Image('projects/ee-indiasat/assets/LULC\_Version2\_Outputs\_NewHierarchy/' +

// block\_name\_lower + '\_temporal\_upd4\_' + y + '-07-01\_' + (y + 1) + '-06-30\_LULCmap\_10m');

img = img.clip(block);

lulc\_imgs.push(img);

}

lulc\_imgs = ee.List(lulc\_imgs);

print('✔️ LULC Loaded:', lulc\_imgs);

// Helper: get season date ranges

function getSeasonDates(season, year) {

year = ee.Number(year); // Ensure it's always an ee.Number

if (season === 'kharif') {

return {

start: ee.Date.fromYMD(year, 7, 1),

end: ee.Date.fromYMD(year, 10, 31)

};

} else if (season === 'rabi') {

return {

start: ee.Date.fromYMD(year, 11, 1),

end: ee.Date.fromYMD(year.add(1), 2, 28)

};

} else {

return {

start: ee.Date.fromYMD(year.add(1), 3, 1),

end: ee.Date.fromYMD(year.add(1), 6, 30)

};

}

}

// Load ET & Runoff

function loadSeasonalETandRunoff(block\_name) {

var years = [2017, 2018, 2019, 2020, 2021, 2022];

var seasons = ['kharif', 'rabi', 'zaid'];

var et\_data = { kharif: [], rabi: [], zaid: [] };

var runoff\_data = { kharif: [], rabi: [], zaid: [] };

seasons.forEach(function(season) {

years.forEach(function(year) {

var dates = getSeasonDates(season, year);

var et\_asset = 'projects/ee-aatif/assets/ET\_MODIS/' + block\_name + '\_MODIS\_ET\_' + season + '\_' + year;

var runoff\_asset = 'projects/ee-aatif/assets/Runoff\_fortnightly/' + block\_name + '\_Runoff\_' + season + '\_' + year;

et\_data[season].push(

ee.Image(et\_asset)

.set({year: year, season: season, 'system:time\_start': dates.start.millis()})

.clip(block)

);

runoff\_data[season].push(

ee.Image(runoff\_asset)

.set({year: year, season: season, 'system:time\_start': dates.start.millis()})

.clip(block)

);

});

});

return {

et: {

kharif: ee.List(et\_data.kharif),

rabi: ee.List(et\_data.rabi),

zaid: ee.List(et\_data.zaid)

},

runoff: {

kharif: ee.List(runoff\_data.kharif),

rabi: ee.List(runoff\_data.rabi),

zaid: ee.List(runoff\_data.zaid)

}

};

}

// Load seasonal SM and rainfall

function loadSeasonalSMandRainfall(season, years) {

return years.map(function(year) {

var dates = getSeasonDates(season, year);

// SM

var sm = ee.ImageCollection("NASA/SMAP/SPL3SMP\_E/005")

.filterBounds(block)

.filterDate(dates.start, dates.end)

.select('soil\_moisture\_am')

.mean()

.multiply(50)

.clip(block)

.set({year: year});

// Rainfall

var rain = ee.ImageCollection("JAXA/GPM\_L3/GSMaP/v6/operational")

.filterDate(dates.start, dates.end)

.filterBounds(block)

.select('hourlyPrecipRate')

.sum()

.clip(block)

.set({year: year});

return { sm: sm, rain: rain };

});

}

// Wrapper

var years = ee.List.sequence(2017, 2022);

var seasonalETRO = loadSeasonalETandRunoff(block\_name);

// SM & Rain for each season

var sm\_rain\_kharif = loadSeasonalSMandRainfall('kharif', years);

var sm\_rain\_rabi = loadSeasonalSMandRainfall('rabi', years);

var sm\_rain\_zaid = loadSeasonalSMandRainfall('zaid', years);

// GW abstraction

function calculateSeasonalGW(etList, smRainList, runoffList, season) {

var indices = ee.List.sequence(0, ee.Number(etList.size()).subtract(1));

var images = indices.map(function(i) {

i = ee.Number(i);

var et = ee.Image(etList.get(i));

var runoff = ee.Image(runoffList.get(i)).divide(900); // to mm

var sm = ee.Image(ee.Dictionary(smRainList.get(i)).get('sm'));

var rain = ee.Image(ee.Dictionary(smRainList.get(i)).get('rain'));

var year = ee.Number(et.get('year'));

var lulc = ee.Image(ee.Algorithms.If(

year.eq(2023), ee.Image(lulc\_imgs.get(5)),

ee.Image(lulc\_imgs.get(year.subtract(2017)))

));

var agriMask = ee.Image(ee.Algorithms.If(

ee.String(season).match('kharif').size().gt(0), lulc.eq(8).or(lulc.eq(10)).or(lulc.eq(11)),

ee.Algorithms.If(

ee.String(season).match('rabi').size().gt(0), lulc.eq(9).or(lulc.eq(10)).or(lulc.eq(11)),

lulc.eq(11)

)

));

var rainMask = rain.gt(0);

var gw\_with\_rain = et.add(sm).subtract(rain.subtract(runoff));

var gw\_no\_rain = et.add(sm);

var gw\_final = gw\_with\_rain.multiply(rainMask)

.add(gw\_no\_rain.multiply(rainMask.not()))

.updateMask(agriMask)

.set({

year: year,

'system:time\_start': et.get('system:time\_start'),

'system:id': block\_name + '\_GW\_' + season + '\_' + year

});

return gw\_final;

});

return ee.ImageCollection.fromImages(images);

}

// Compute seasonal GW

var gw\_kharif = calculateSeasonalGW(seasonalETRO.et.kharif, sm\_rain\_kharif, seasonalETRO.runoff.kharif, 'kharif');

var gw\_rabi = calculateSeasonalGW(seasonalETRO.et.rabi, sm\_rain\_rabi, seasonalETRO.runoff.rabi, 'rabi');

var gw\_zaid = calculateSeasonalGW(seasonalETRO.et.zaid, sm\_rain\_zaid, seasonalETRO.runoff.zaid, 'zaid');

// Summary by year

function summarize(gw, season) {

return ee.ImageCollection.fromImages(years.map(function(y) {

var img = gw.filter(ee.Filter.eq('year', y)).reduce(ee.Reducer.sum());

return img.set({

year: y,

'system:id': block\_name + '\_GW\_' + season + '\_sum\_' + y

});

}));

}

var gw\_kharif\_yearly = summarize(gw\_kharif, 'kharif');

var gw\_rabi\_yearly = summarize(gw\_rabi, 'rabi');

var gw\_zaid\_yearly = summarize(gw\_zaid, 'zaid');

// 🌍 Visualization

var viz = {min: 0, max: 300, palette: ['#ffffcc', '#a1dab4', '#41b6c4', '#2c7fb8', '#253494']};

Map.centerObject(block, 10);

Map.addLayer(gw\_kharif\_yearly.reduce(ee.Reducer.mean()), viz, 'Mean Kharif GW');

Map.addLayer(gw\_rabi\_yearly.reduce(ee.Reducer.mean()), viz, 'Mean Rabi GW');

Map.addLayer(gw\_zaid\_yearly.reduce(ee.Reducer.mean()), viz, 'Mean Zaid GW');

Map.addLayer(block.style({color: 'black', fillColor: '00000000', width: 2}), {}, 'Block');

// Export helper function

function exportSeasonToDrive(gwImages, season) {

var yearsList = ee.List.sequence(2017, 2022);

// Export yearly images

yearsList.evaluate(function(yrs) {

yrs.forEach(function(y) {

var image = gwImages.filter(ee.Filter.eq('year', y)).first();

// Export.image.toDrive({

// image: image,

// description: 'GW\_' + block\_name + '\_' + season + '\_' + y,

// fileNamePrefix: 'GW\_' + block\_name + '\_' + season + '\_' + y,

// folder: 'Equity\_Aatif', // You can change this folder name

// region: block.geometry(),

// scale: 30,

// maxPixels: 1e9,

// fileFormat: 'GeoTIFF'

// });

});

});

// Export multi-year mean image

Export.image.toDrive({

image: gwImages.reduce(ee.Reducer.mean()),

description: 'GW\_' + block\_name + '\_' + season + '\_Mean\_2017\_2022',

fileNamePrefix: 'GW\_' + block\_name + '\_' + season + '\_Mean\_2017\_2022',

folder: 'Equity\_Aatif',

region: block.geometry(),

scale: 30,

maxPixels: 1e9,

fileFormat: 'GeoTIFF'

});

}

exportSeasonToDrive(gw\_kharif\_yearly, 'kharif');

exportSeasonToDrive(gw\_rabi\_yearly, 'rabi');

exportSeasonToDrive(gw\_zaid\_yearly, 'zaid');

// // Print results for all seasons

// print('Yearly Kharif GW Abstracted on Agricultural Pixels:', gw\_kharif\_yearly);

// print('Mean Kharif GW Abstracted (2017-2022):', gw\_kharif\_yearly.reduce(ee.Reducer.mean()));

// print('Yearly Rabi GW Abstracted on Agricultural Pixels:', gw\_rabi\_yearly);

// print('Mean Rabi GW Abstracted (2017-2022):', gw\_rabi\_yearly.reduce(ee.Reducer.mean()));

// print('Yearly Zaid GW Abstracted on Agricultural Pixels:', gw\_zaid\_yearly);

// print('Mean Zaid GW Abstracted (2017-2022):', gw\_zaid\_yearly.reduce(ee.Reducer.mean()));