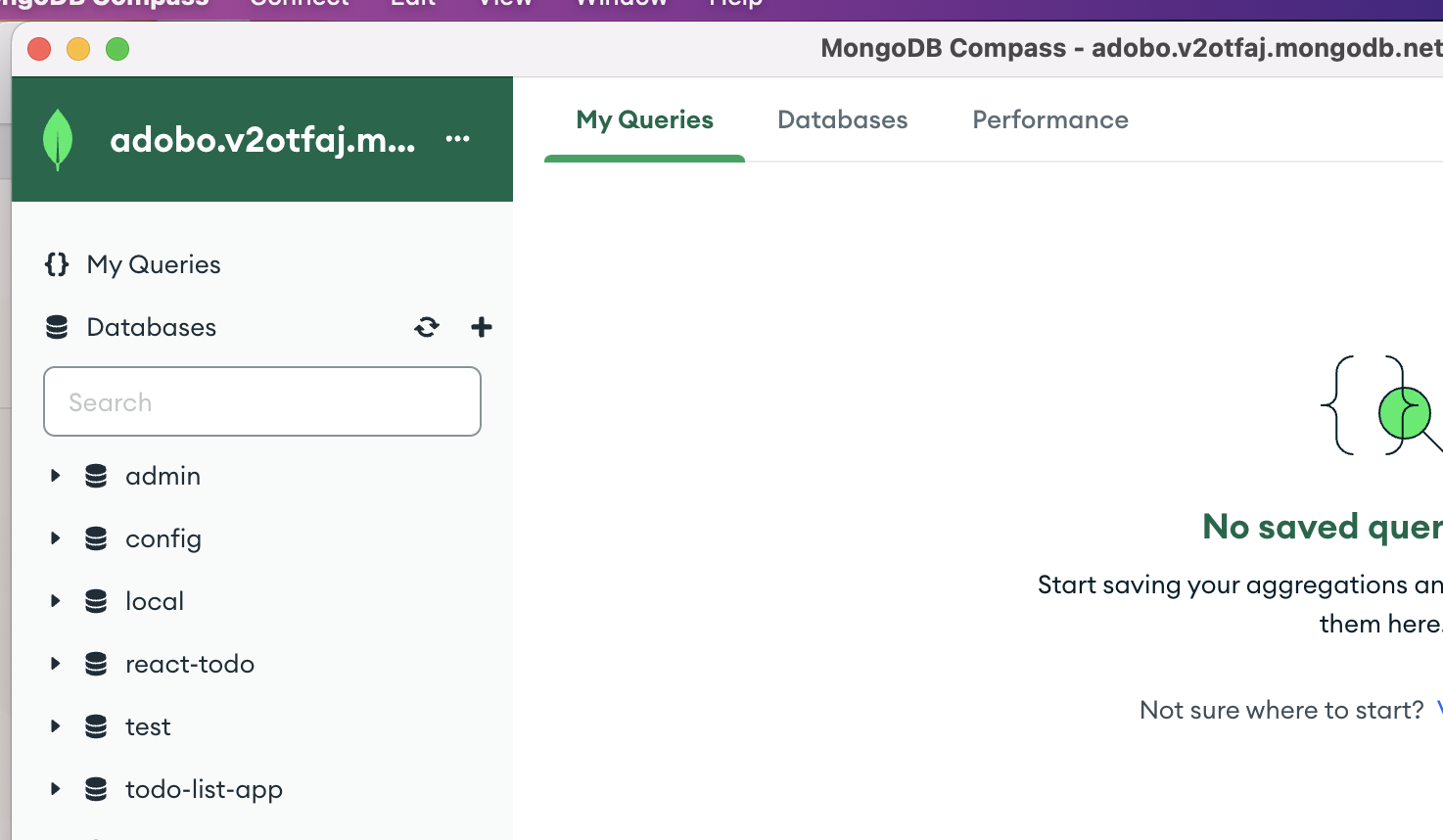
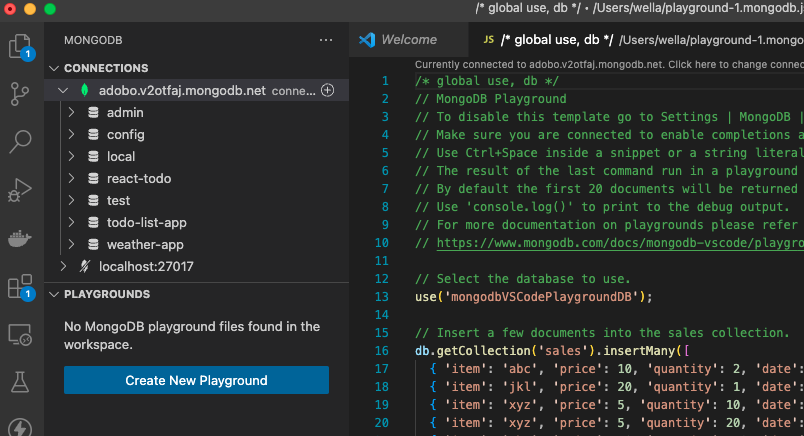
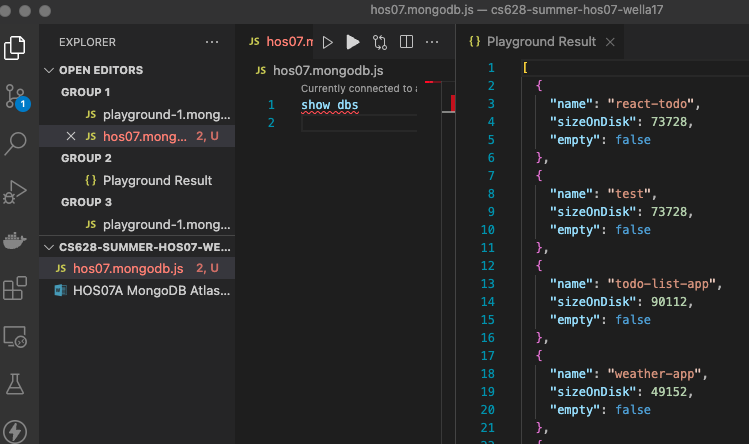
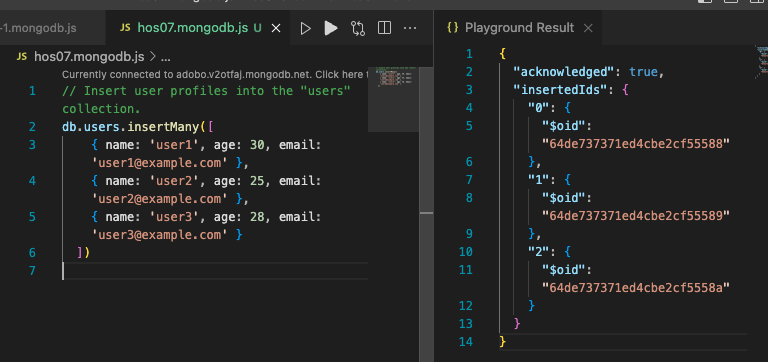
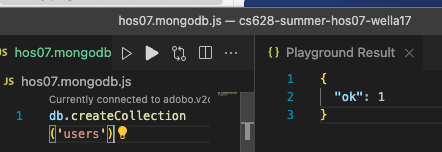
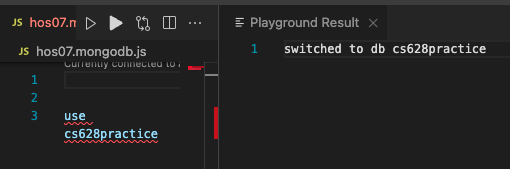
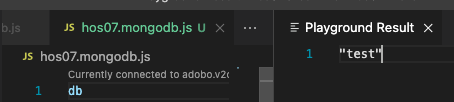
CS628 Full-Stack Development – Web App

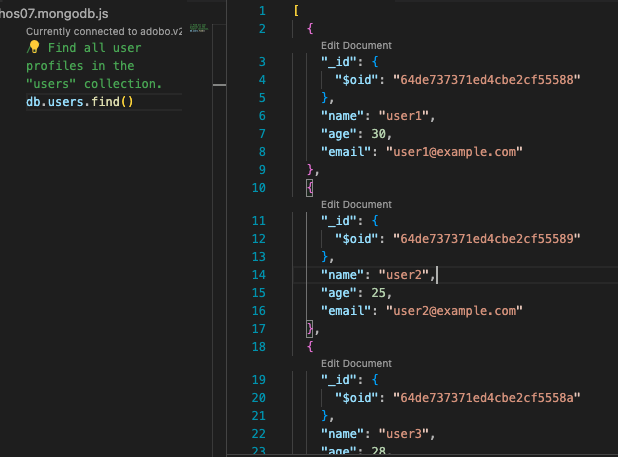
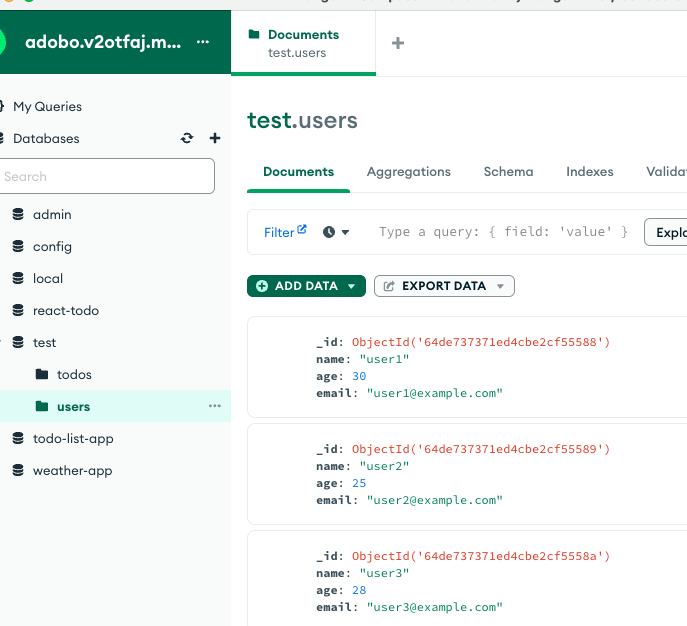
**HOS07A: MongoDB Atlas**

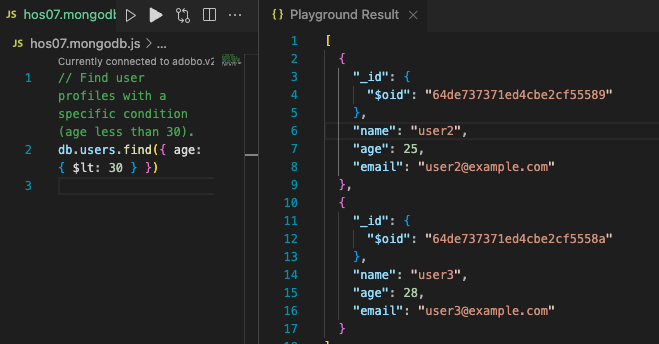
**Section 4: Sign up for MongoDB Atlas**

**Section 6: Setting up environment - MongoDB for VS Code on GitHub Codespace**

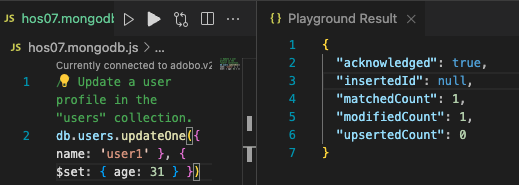
**Section 7: Learn MongoDB step by step.**

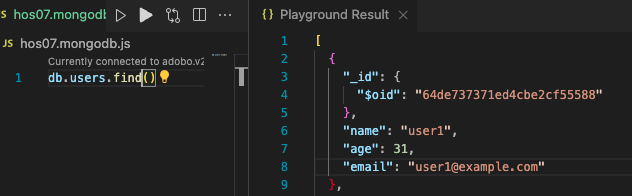
****

****

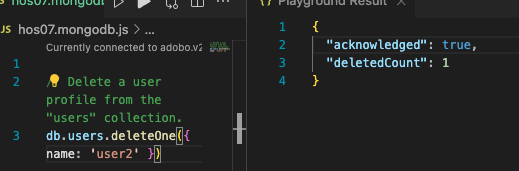
****

**// Update a user profile in the "users" collection.**

**db.users.updateOne({ name: 'user1' }, { $set: { age: 31 } })**

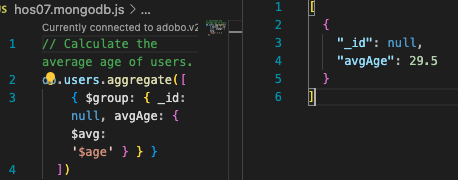
db.users.find()

// Delete a user profile from the "users" collection.

db.users.deleteOne({ name: 'user2' })

// Calculate the average age of users.

db.users.aggregate([

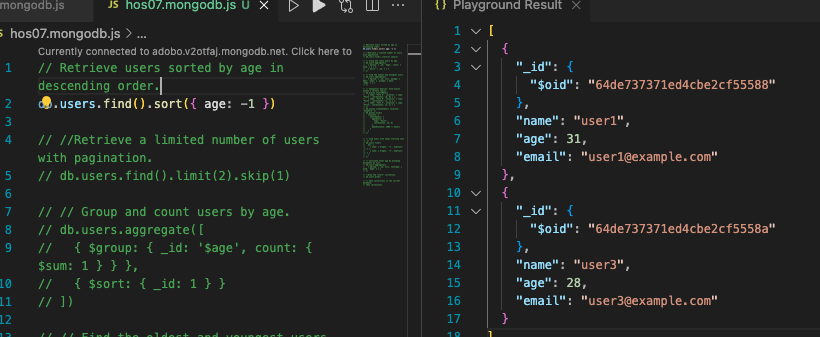
{ $group: { \_id: null, avgAge: { $avg: '$age' } } }

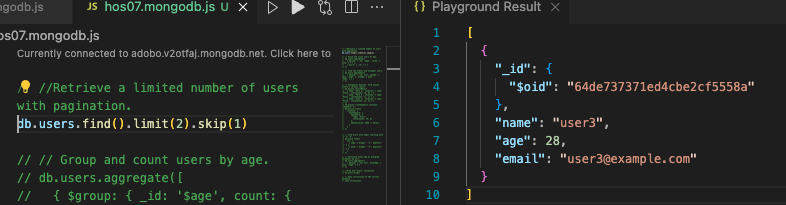
])

// Create an index on the "email" field for faster searches.

db.users.createIndex({ email: 1 })

// Retrieve users sorted by age in descending order.

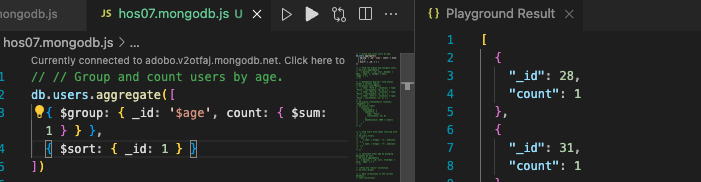
db.users.find().sort({ age: -1 })

db.users.find().limit(2).skip(1)

// Group and count users by age.

db.users.aggregate([

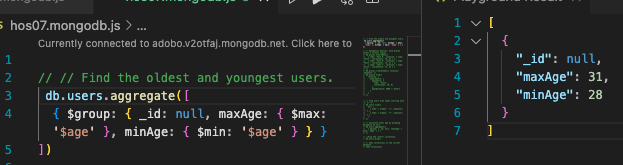
{ $group: { \_id: '$age', count: { $sum: 1 } } },

{ $sort: { \_id: 1 } }

])

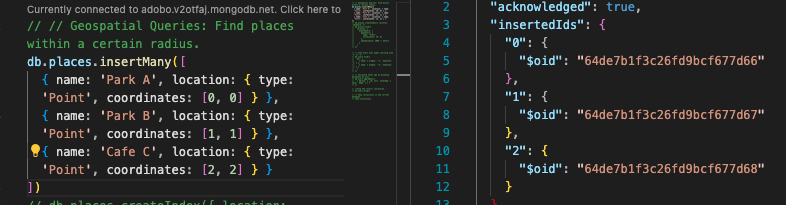
db.users.aggregate([

{ $group: { \_id: null, maxAge: { $max: '$age' }, minAge: { $min: '$age' } } }

])

db.users.aggregate([

{ $group: { \_id: null, maxAge: { $max: '$age' }, minAge: { $min: '$age' } } }

])

db.places.find({

location: {

$nearSphere: {

$geometry: {

type: 'Point',

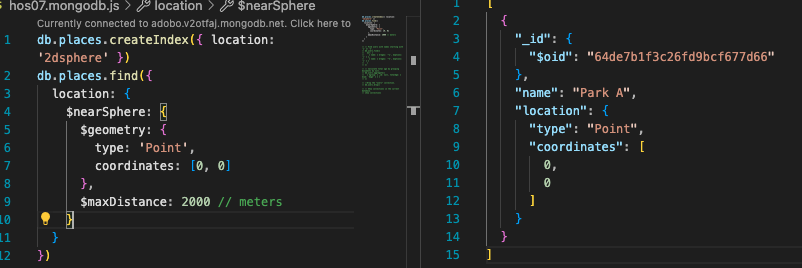
coordinates: [0, 0]

},

$maxDistance: 2000 // meters

}

}

})

// Find users with names starting with 'u' or 'C'.

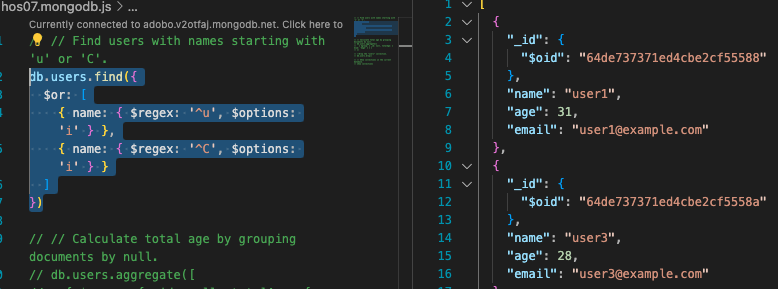
db.users.find({

$or: [

{ name: { $regex: '^u', $options: 'i' } },

{ name: { $regex: '^C', $options: 'i' } }

]

})

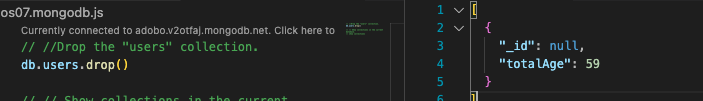
// Calculate total age by grouping documents by null.

db.users.aggregate([

{ $group: { \_id: null, totalAge: { $sum: '$age' } } }

])

//Drop the "users" collection.

db.users.drop()

// Show collections in the current database.

show collections