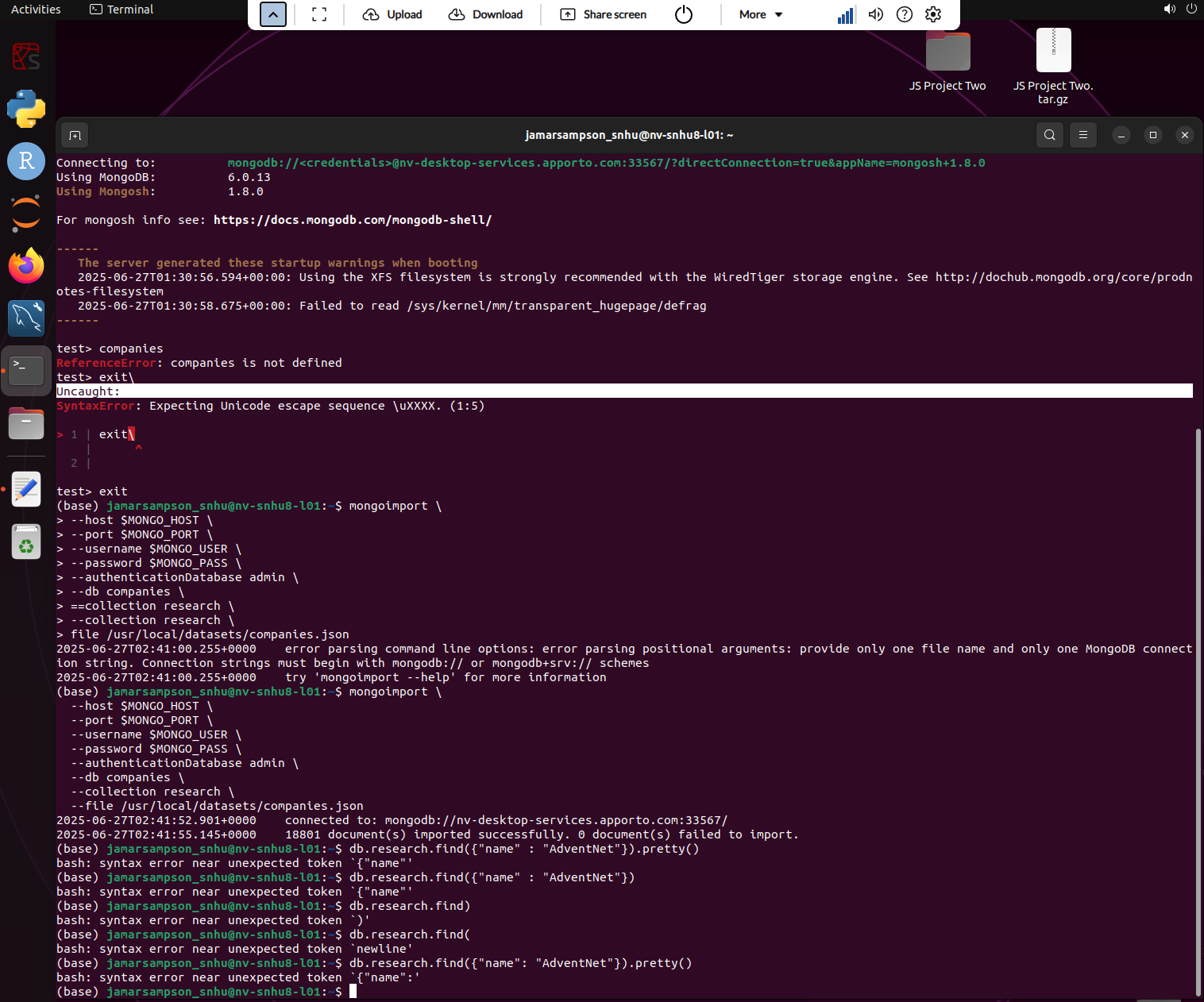
The following document aims to be used as an illustration for leveraging MongoDB’s advanced query capabilities, enabling more sophisticated data analysis directly on the database level. MongoDB's aggregation pipeline offers a suite of operators designed to perform set theory operations, including grouping and intersections across collections.

The command below was used to import the data into the MongoDB database.

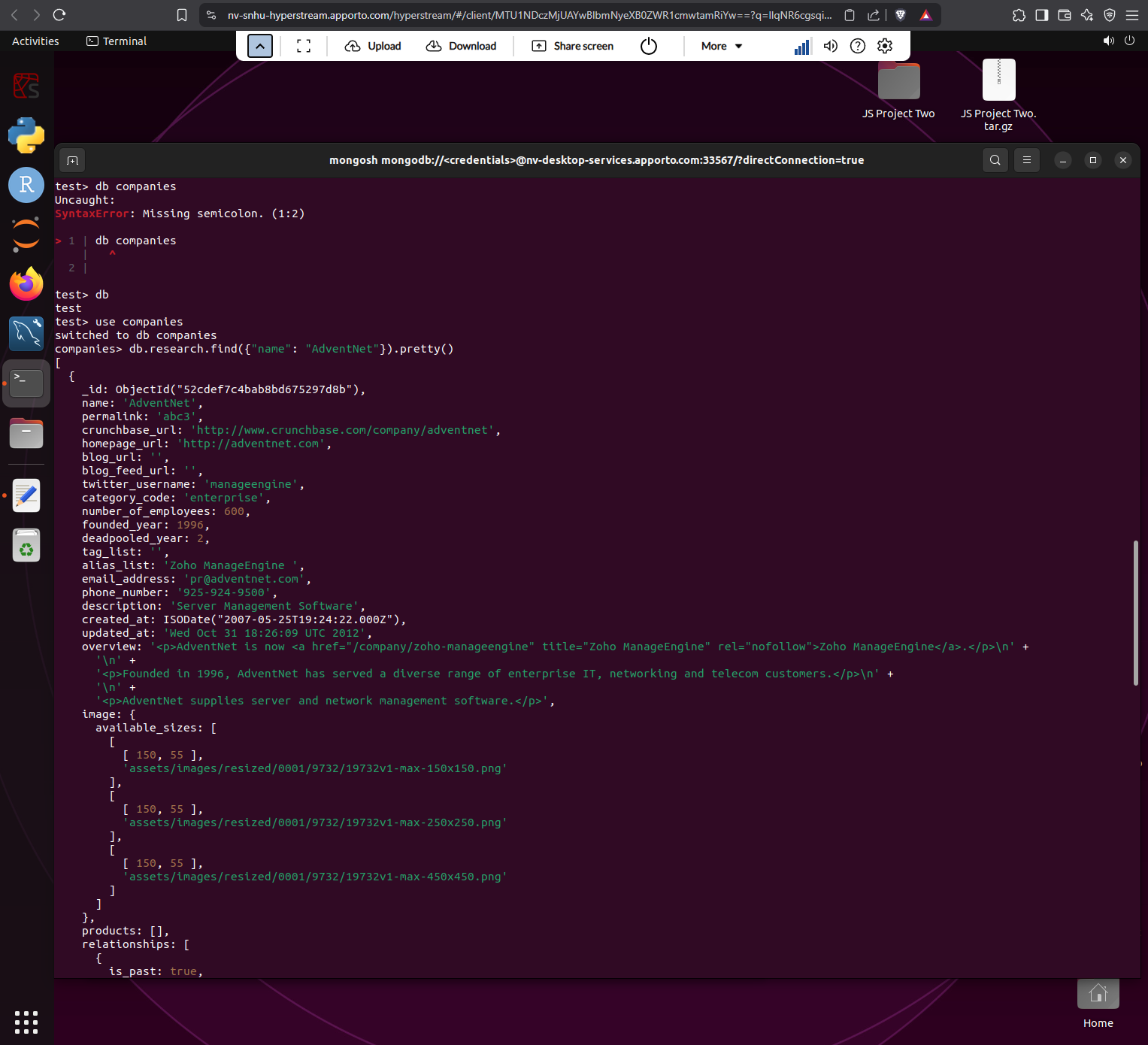
mongoimport \ --host $MONGO\_HOST \ --port MONGO\_PORT \ --username $MONGO\_USER \ --password MONGO\_PASS \ --authenticationDatabase admin \ --db companies \ --collection research \ --file /usr/local/datasets/companies.json



This allowed the database to connect & successfully import the documents.

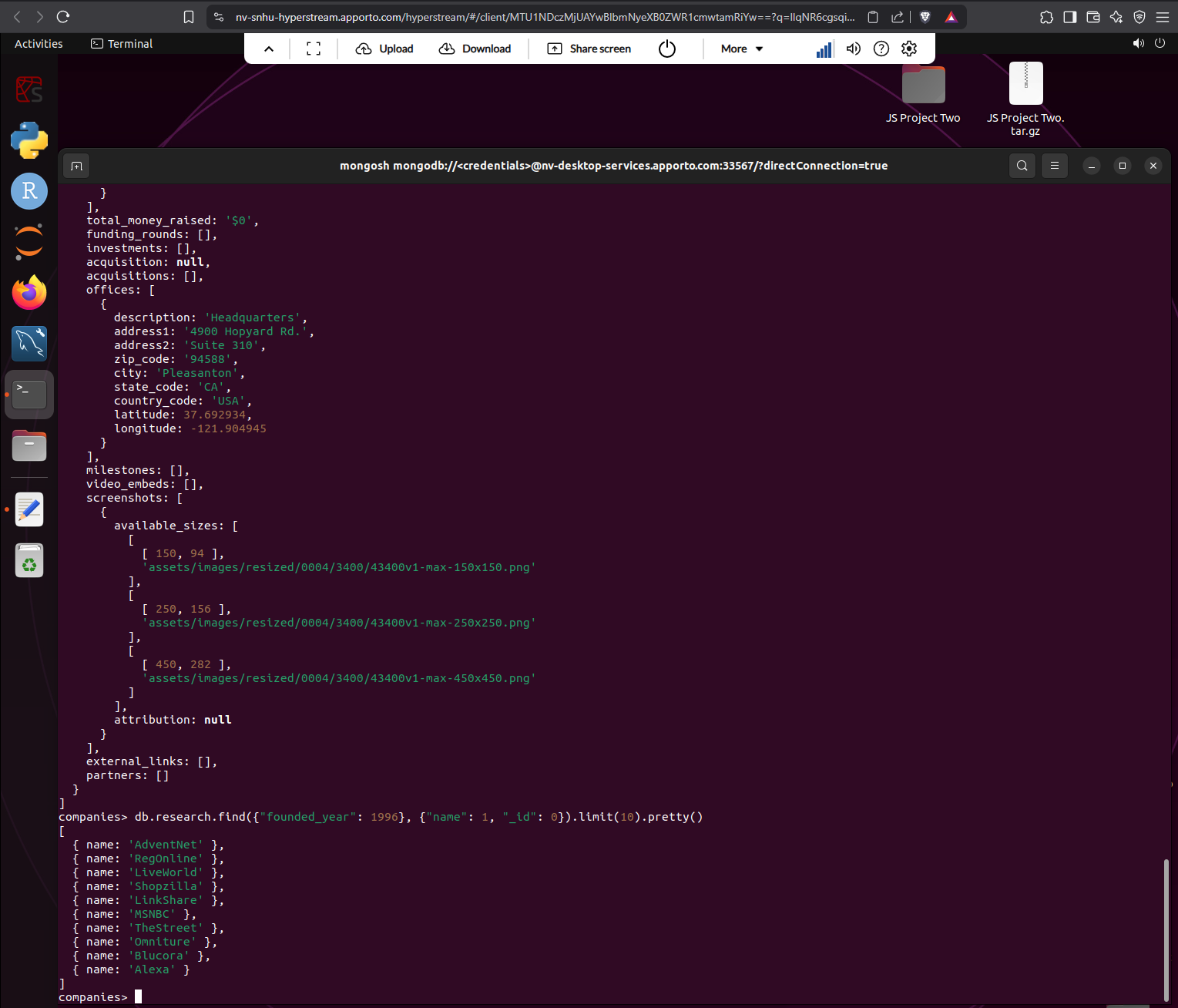
In order to verify the success of the data import, I ran the following query to check for the company “AdventNet”.

db.research.find({"name" : "AdventNet"}).pretty()



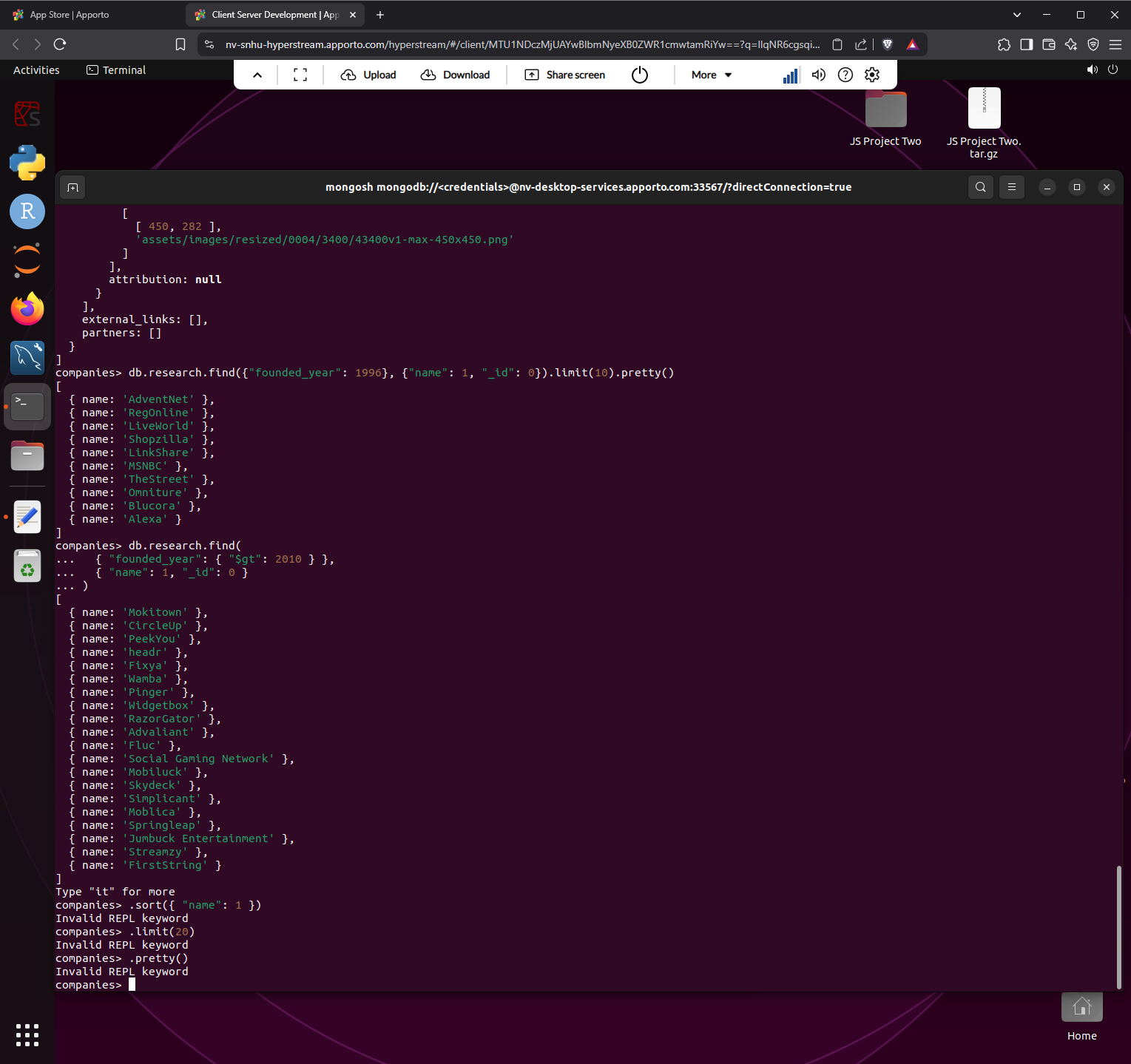
An additional query was run to find companies founded in 1996, limited to displaying just the name field, & no more than 10 results.

db.research.find({"founded\_year" : 1996}, {"name" : 1}).limit(10).pretty()



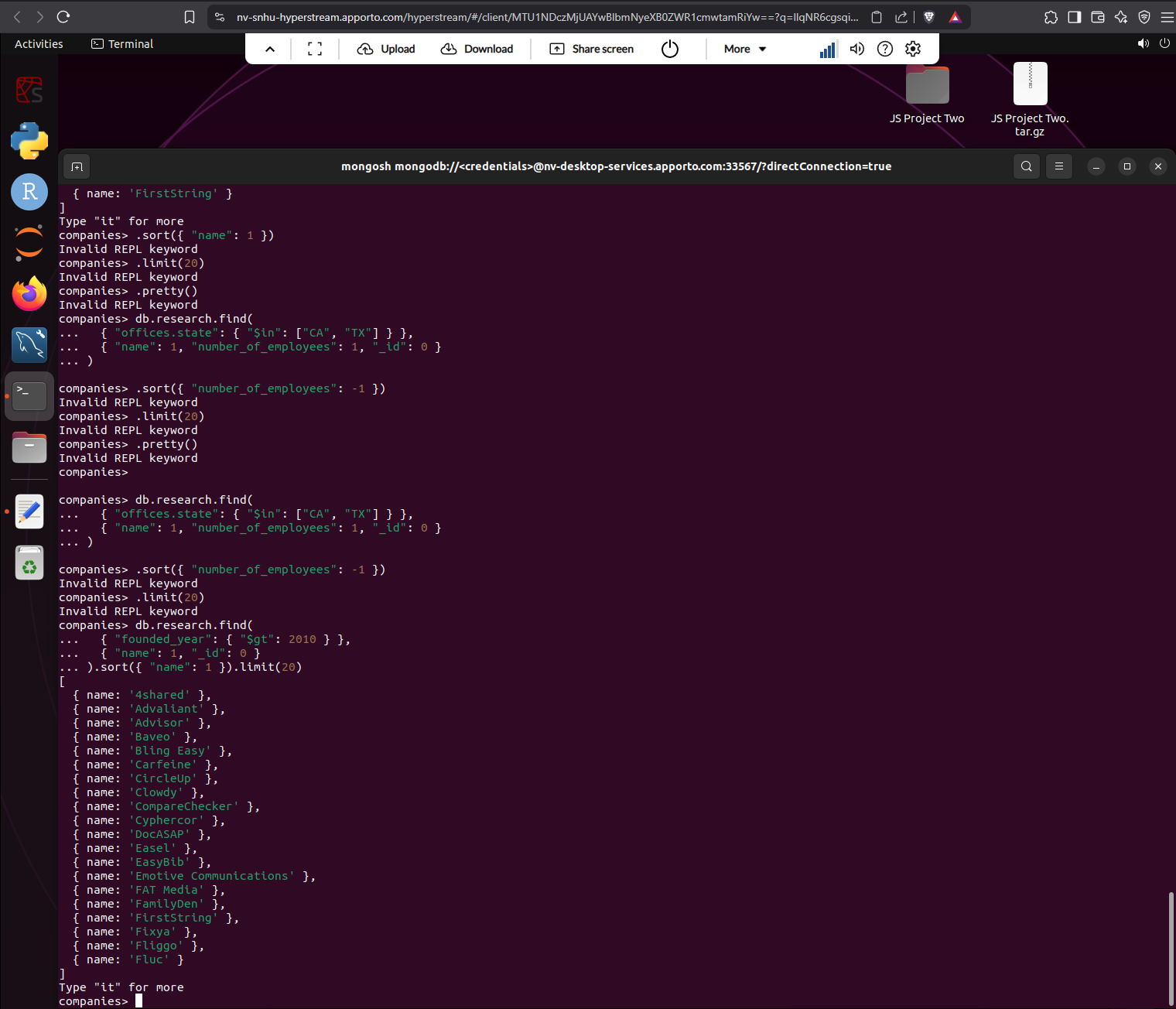
An additional query was run to find companies founded after 2010, limited to displaying just the name field, & no more than 20 results.

db.research.find({"founded\_year": {$gt: 2010}}, {"name": 1, "\_id": 0}).sort({"name": 1}).limit(20).pretty()



An additional query was run to find the names of 20 companies with offices in either California or Texas, ordered by the number of employees and sorted largest to smallest.

db.research.find({"offices.state\_code": {$in: ["CA", "TX"]}}, {"name": 1, "number\_of\_employees": 1, "\_id": 0}).sort({"number\_of\_employees": -1}).limit(20).pretty()



Lastly, I was tasked with designing an aggregation pipeline to show the total number of offices by state for all the companies that have offices in the United States.

db.research.aggregate([  
 { $unwind: "$offices" },  
 { $match: { "offices.country\_code": "USA" } },  
 { $group: { \_id: "$offices.state\_code", totalOffices: { $sum: 1 } } },  
 { $sort: { totalOffices: -1 } }  
])

