**Experiment 10:**

Develop an aggregation pipeline to illustrate Text search on Catalog data collection.

db.catalog.insertMany([

{

\_id: 1,

name: "Wireless Mouse",

description: "A compact wireless mouse with ergonomic design and long battery life.",

category: "Electronics",

price: 499

},

{

\_id: 2,

name: "Bluetooth Speaker",

description: "Portable speaker with powerful bass and Bluetooth connectivity.",

category: "Electronics",

price: 1199

},

{

\_id: 3,

name: "Office Chair",

description: "Ergonomic office chair with lumbar support and mesh backrest.",

category: "Furniture",

price: 2999

},

{

\_id: 4,

name: "Wireless Earbuds",

description: "Noise-cancelling wireless earbuds with charging case.",

category: "Electronics",

price: 1799

}

])

db.catalog.createIndex({ name: "text", description: "text" , category: “text”})

To search for items related to "wireless", sort them by price, and return only the name and price:

db.catalog.aggregate([

{

$match: {

$text: { $search: "wireless" }

}

},

{

$sort: { score: { $meta: "textScore" } }

},

{

$project: {

\_id: 0,

name: 1,

price: 1,

score: { $meta: "textScore" }

}

}

])