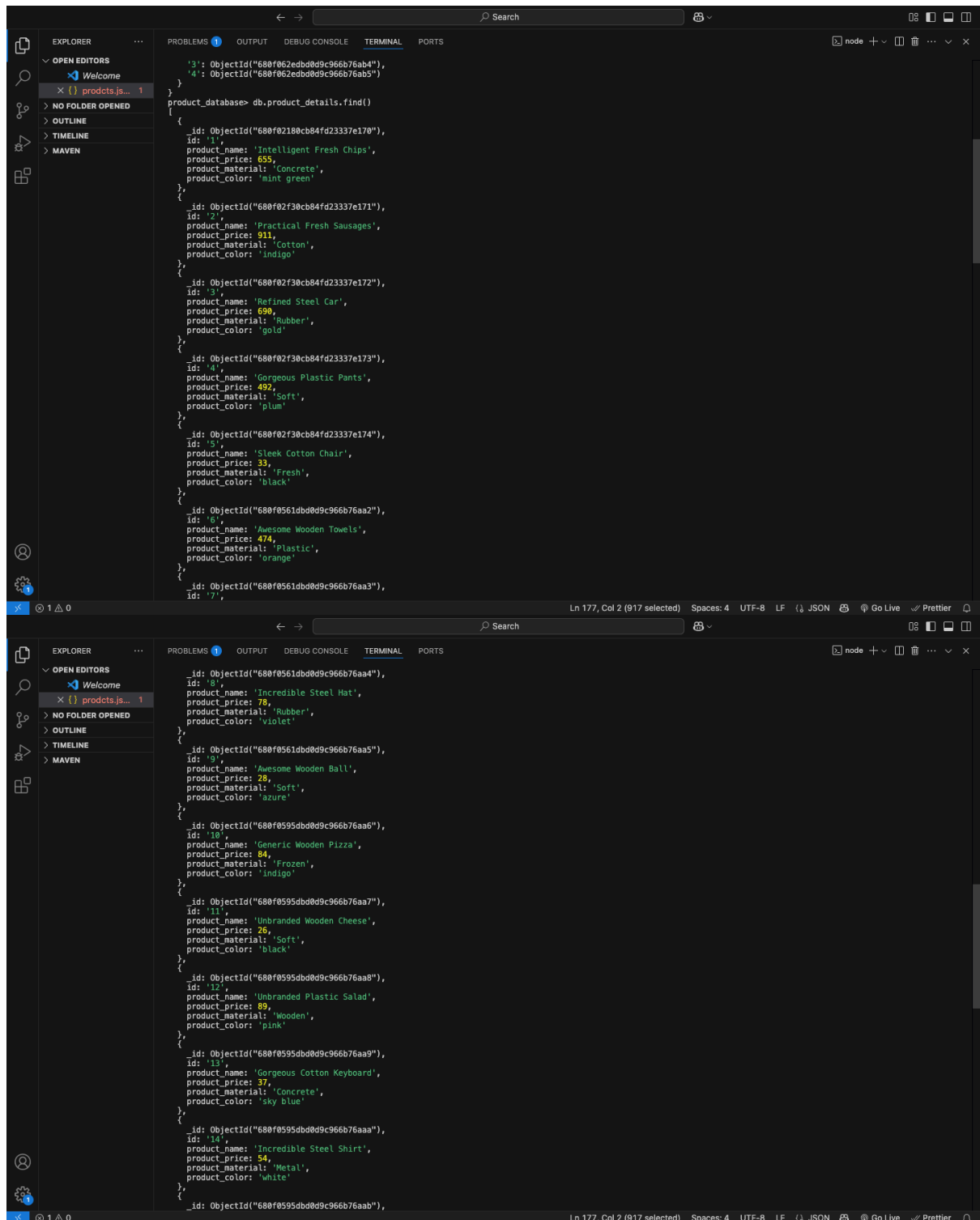


MongoDB Task

1. Find all the information about each products?

Command: db.product_details.find()

Screen Shot of output:



```
product_database> db.product_details.find()
{
  '_id': ObjectId('680f02180cb84fd23337e170'),
  'id': '1',
  'product_name': 'Intelligent Fresh Chips',
  'product_price': 655,
  'product_material': 'Concrete',
  'product_color': 'mint green'
},
{
  '_id': ObjectId('680f02f30cb84fd23337e171'),
  'id': '2',
  'product_name': 'Practical Fresh Sausages',
  'product_price': 911,
  'product_material': 'Cotton',
  'product_color': 'indigo'
},
{
  '_id': ObjectId('680f02f30cb84fd23337e172'),
  'id': '3',
  'product_name': 'Refined Steel Car',
  'product_price': 690,
  'product_material': 'Rubber',
  'product_color': 'gold'
},
{
  '_id': ObjectId('680f02f30cb84fd23337e173'),
  'id': '4',
  'product_name': 'Gorgeous Plastic Pants',
  'product_price': 492,
  'product_material': 'Soft',
  'product_color': 'plum'
},
{
  '_id': ObjectId('680f02f30cb84fd23337e174'),
  'id': '5',
  'product_name': 'Sleek Cotton Chair',
  'product_price': 33,
  'product_material': 'Fresh',
  'product_color': 'black'
},
{
  '_id': ObjectId('680f0561dbd8d9c966b76aa2'),
  'id': '6',
  'product_name': 'Awesome Wooden Towels',
  'product_price': 474,
  'product_material': 'Plastic',
  'product_color': 'orange'
},
{
  '_id': ObjectId('680f0561dbd8d9c966b76aa3'),
  'id': '7',
  'product_name': 'Incredible Steel Hat',
  'product_price': 78,
  'product_material': 'Rubber',
  'product_color': 'violet'
},
{
  '_id': ObjectId('680f0561dbd8d9c966b76aa5'),
  'id': '8',
  'product_name': 'Awesome Wooden Ball',
  'product_price': 28,
  'product_material': 'Soft',
  'product_color': 'azure'
},
{
  '_id': ObjectId('680f0595dbd8d9c966b76aa6'),
  'id': '10',
  'product_name': 'Generic Wooden Pizza',
  'product_price': 84,
  'product_material': 'Frozen',
  'product_color': 'indigo'
},
{
  '_id': ObjectId('680f0595dbd8d9c966b76aa7'),
  'id': '11',
  'product_name': 'Unbranded Wooden Cheese',
  'product_price': 26,
  'product_material': 'Soft',
  'product_color': 'black'
},
{
  '_id': ObjectId('680f0595dbd8d9c966b76aa8'),
  'id': '12',
  'product_name': 'Unbranded Plastic Salad',
  'product_price': 89,
  'product_material': 'Wooden',
  'product_color': 'pink'
},
{
  '_id': ObjectId('680f0595dbd8d9c966b76aa9'),
  'id': '13',
  'product_name': 'Gorgeous Cotton Keyboard',
  'product_price': 37,
  'product_material': 'Concrete',
  'product_color': 'sky blue'
},
{
  '_id': ObjectId('680f0595dbd8d9c966b76aaa'),
  'id': '14',
  'product_name': 'Incredible Steel Shirt',
  'product_price': 54,
  'product_material': 'Metal',
  'product_color': 'white'
},
{
  '_id': ObjectId('680f0595dbd8d9c966b76aab'),
  'id': '15',
  'product_name': 'Incredible Steel Hat',
  'product_price': 78,
  'product_material': 'Rubber',
  'product_color': 'violet'
}
```

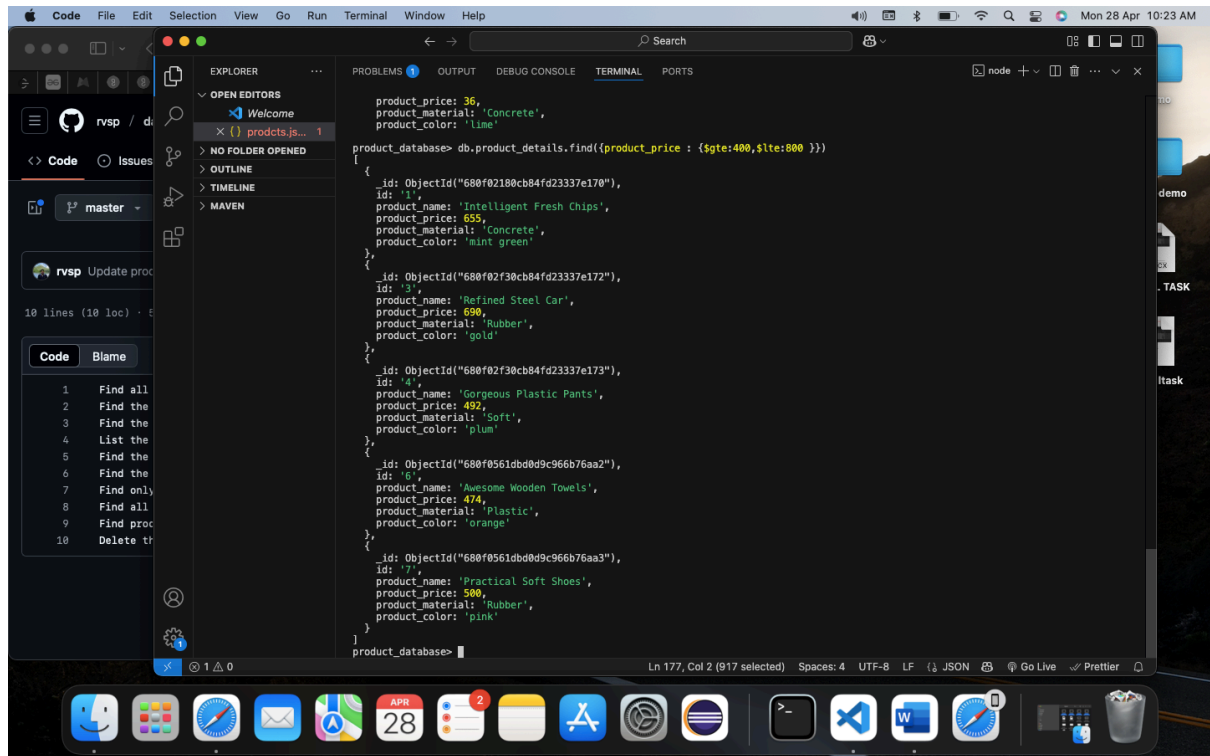
```
    {
      _id: ObjectId("680f0595dbd8d9c966b76aab"),
      id: '15',
      product_name: 'Ergonomic Cotton Hat',
      product_price: 43,
      product_material: 'Rubber',
      product_color: 'mint green'
    },
    {
      _id: ObjectId("680f05f4dbd8d9c966b76aad"),
      id: '16',
      product_name: 'Small Soft Chair',
      product_price: 47,
      product_material: 'Cotton',
      product_color: 'teal'
    },
    {
      _id: ObjectId("680f05f4dbd8d9c966b76aad"),
      id: '17',
      product_name: 'Incredible Metal Car',
      product_price: 36,
      product_material: 'Fresh',
      product_color: 'indigo'
    },
    {
      _id: ObjectId("680f05f4dbd8d9c966b76aaf"),
      id: '18',
      product_name: 'Licensed Plastic Bacon',
      product_price: 88,
      product_material: 'Steel',
      product_color: 'yellow'
    },
    {
      _id: ObjectId("680f05f4dbd8d9c966b76abf"),
      id: '19',
      product_name: 'Intelligent Cotton Chips',
      product_price: 46,
      product_material: 'Soft',
      product_color: 'azure'
    },
    {
      _id: ObjectId("680f05f4dbd8d9c966b76ab0"),
      id: '20',
      product_name: 'Handcrafted Wooden Bacon',
      product_price: 36,
      product_material: 'Concrete',
      product_color: 'lime'
    }
  ]
  Type "it" for more
  product_database> it
  {
    _id: ObjectId("680f062edbd8d9c966b76ab1"),
    id: '21',
    product_name: 'Unbranded Granite Chicken',
    product_price: 90,
    product_material: 'Metal',
    product_color: 'gold'
```

```
    {
      _id: ObjectId("680f062edbd8d9c966b76ab1"),
      id: '21',
      product_name: 'Unbranded Granite Chicken',
      product_price: 90,
      product_material: 'Metal',
      product_color: 'gold'
    },
    {
      _id: ObjectId("680f062edbd8d9c966b76ab2"),
      id: '22',
      product_name: 'Ergonomic Soft Hat',
      product_price: 99,
      product_material: 'Rubber',
      product_color: 'black'
    },
    {
      _id: ObjectId("680f062edbd8d9c966b76ab3"),
      id: '23',
      product_name: 'Intelligent Steel Pizza',
      product_price: 95,
      product_material: 'Cotton',
      product_color: 'azure'
    },
    {
      _id: ObjectId("680f062edbd8d9c966b76ab4"),
      id: '24',
      product_name: 'Tasty Rubber Cheese',
      product_price: 47,
      product_material: 'Frozen',
      product_color: 'orchid'
    },
    {
      _id: ObjectId("680f062edbd8d9c966b76ab5"),
      id: '25',
      product_name: 'Licensed Steel Car',
      product_price: 20,
      product_material: 'Cotton',
      product_color: 'indigo'
    }
  ]
  product_database> 
```

2. Find the product price which are between 400 to 800?

Command: db.product_details.find({product_price : {\$gte:400,\$lte:800 }})

Screen Shot of output:



The screenshot shows a VS Code window with a terminal running a MongoDB query. The query is `product_database> db.product_details.find({product_price : {$gte:400,$lte:800 }})`. The output displays a JSON array of product details. The first product is 'Intelligent Fresh Chips' with a price of 655. The second is 'Refined Steel Car' with a price of 690. The third is 'Gorgeous Plastic Pants' with a price of 492. The fourth is 'Awesome Wooden Towels' with a price of 474. The fifth is 'Practical Soft Shoes' with a price of 500. The terminal status bar at the bottom indicates 'Ln 177, Col 2 (917 selected) Spaces: 4 UTF-8 LF (JSON) Go Live Prettier'.

```
product_price: 36,
product_material: 'Concrete',
product_color: 'lime'

product_database> db.product_details.find({product_price : {$gte:400,$lte:800 }})
{
  _id: ObjectId("680f02180cb84fd23337e170"),
  id: '1',
  product_name: 'Intelligent Fresh Chips',
  product_price: 655,
  product_material: 'Concrete',
  product_color: 'mint green'
},
{
  _id: ObjectId("680f02f38cb84fd23337e172"),
  id: '3',
  product_name: 'Refined Steel Car',
  product_price: 690,
  product_material: 'Rubber',
  product_color: 'gold'
},
{
  _id: ObjectId("680f02f38cb84fd23337e173"),
  id: '4',
  product_name: 'Gorgeous Plastic Pants',
  product_price: 492,
  product_material: 'Soft',
  product_color: 'plum'
},
{
  _id: ObjectId("680f0561dbd0d9c966b76aa2"),
  id: '6',
  product_name: 'Awesome Wooden Towels',
  product_price: 474,
  product_material: 'Plastic',
  product_color: 'orange'
},
{
  _id: ObjectId("680f0561dbd0d9c966b76aa3"),
  id: '7',
  product_name: 'Practical Soft Shoes',
  product_price: 500,
  product_material: 'Rubber',
  product_color: 'pink'
}
}

product_database>
```

3. Find the product price which are not between 400 to 600?

```
Command: db.product_details.find({
  $or:[
    {product_price:{$lt: 400 }},
    {product_price:{$gt: 600 }}
  ]
})
```

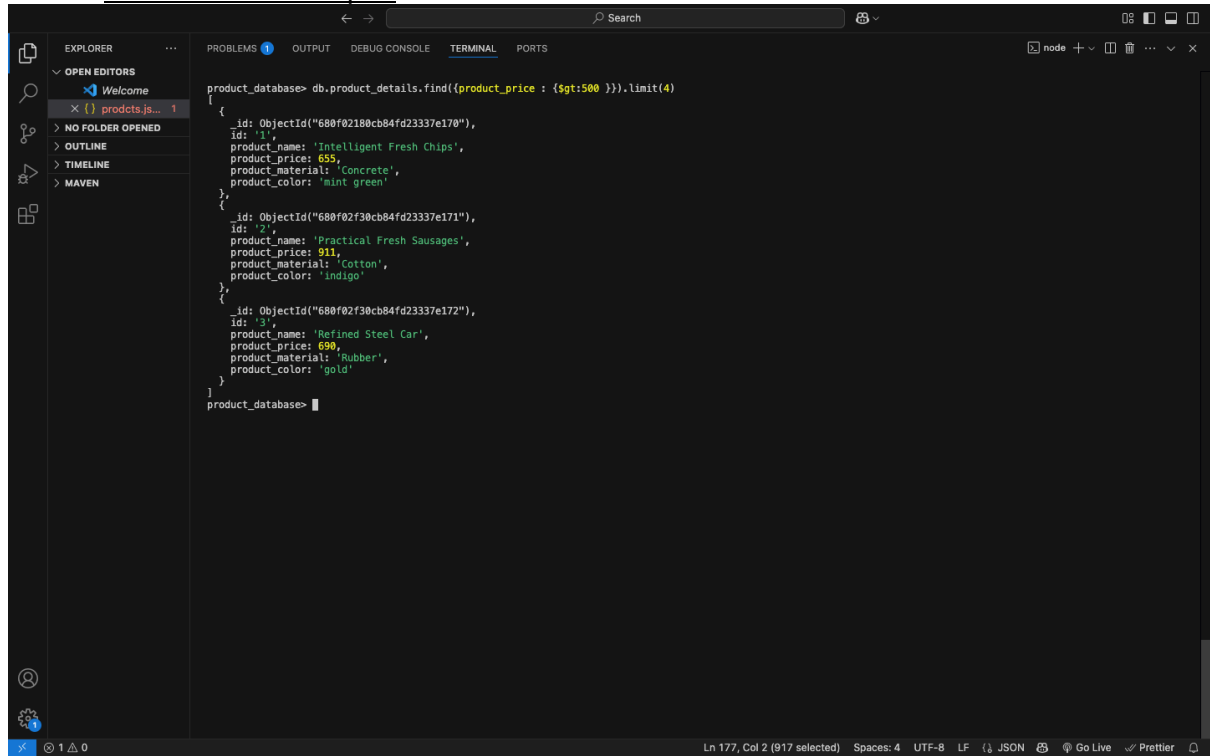
Screen Shot of output:

```
product_database> db.product_details.find({$or:[{product_price:{$lt: 400 }},{product_price:{$gt: 600 }}}})
{
  _id: ObjectId("680f02180cb84fd23337e170"),
  id: '1',
  product_name: 'Intelligent Fresh Chips',
  product_price: 655,
  product_material: 'Concrete',
  product_color: 'mint green'
},
{
  _id: ObjectId("680f02f30cb84fd23337e171"),
  id: '2',
  product_name: 'Practical Fresh Sausages',
  product_price: 911,
  product_material: 'Cotton',
  product_color: 'indigo'
},
{
  _id: ObjectId("680f02f30cb84fd23337e172"),
  id: '3',
  product_name: 'Refined Steel Car',
  product_price: 690,
  product_material: 'Rubber',
  product_color: 'gold'
},
{
  _id: ObjectId("680f02f30cb84fd23337e174"),
  id: '5',
  product_name: 'Sleek Cotton Chair',
  product_price: 33,
  product_material: 'Fresh',
  product_color: 'black'
},
{
  _id: ObjectId("680f0561dbd0d9c966b76aa4"),
  id: '8',
  product_name: 'Incredible Steel Hat',
  product_price: 78,
  product_material: 'Rubber',
  product_color: 'violet'
},
{
  _id: ObjectId("680f0561dbd0d9c966b76aa5"),
  id: '9',
  product_name: 'Awesome Wooden Ball',
  product_price: 29,
  product_material: 'Soft',
  product_color: 'azure'
},
{
  _id: ObjectId("680f0595dbd0d9c966b76aa6"),
  id: '10',
  product_name: 'Generic Wooden Pizza',
  product_price: 94,
  product_material: 'Frozen',
  product_color: 'indigo'
},
{
  _id: ObjectId("680f05f4dbd0d9c966b76aae"),
  id: '18',
  product_name: 'Licensed Plastic Bacon',
  product_price: 88,
  product_material: 'Steel',
  product_color: 'yellow'
},
{
  _id: ObjectId("680f05f4dbd0d9c966b76aaf"),
  id: '19',
  product_name: 'Intelligent Cotton Chips',
  product_price: 46,
  product_material: 'Soft',
  product_color: 'azure'
},
{
  _id: ObjectId("680f05f4dbd0d9c966b76ab0"),
  id: '20',
  product_name: 'Handcrafted Wooden Bacon',
  product_price: 36,
  product_material: 'Concrete',
  product_color: 'lime'
},
{
  _id: ObjectId("680f062edbd0d9c966b76ab1"),
  id: '21',
  product_name: 'Unbranded Granite Chicken',
  product_price: 90,
  product_material: 'Metal',
  product_color: 'gold'
},
{
  _id: ObjectId("680f062edbd0d9c966b76ab2"),
  id: '22',
  product_name: 'Ergonomic Soft Hat',
  product_price: 99,
  product_material: 'Rubber',
  product_color: 'black'
},
{
  _id: ObjectId("680f062edbd0d9c966b76ab3"),
  id: '23',
  product_name: 'Intelligent Steel Pizza',
  product_price: 95,
  product_material: 'Cotton',
  product_color: 'azure'
}
```

4. List the four product which are greater than 500 in price ?

Command: db.product_details.find({product_price : {\$gt:500 }}).limit(4)

Screen Shot of output:



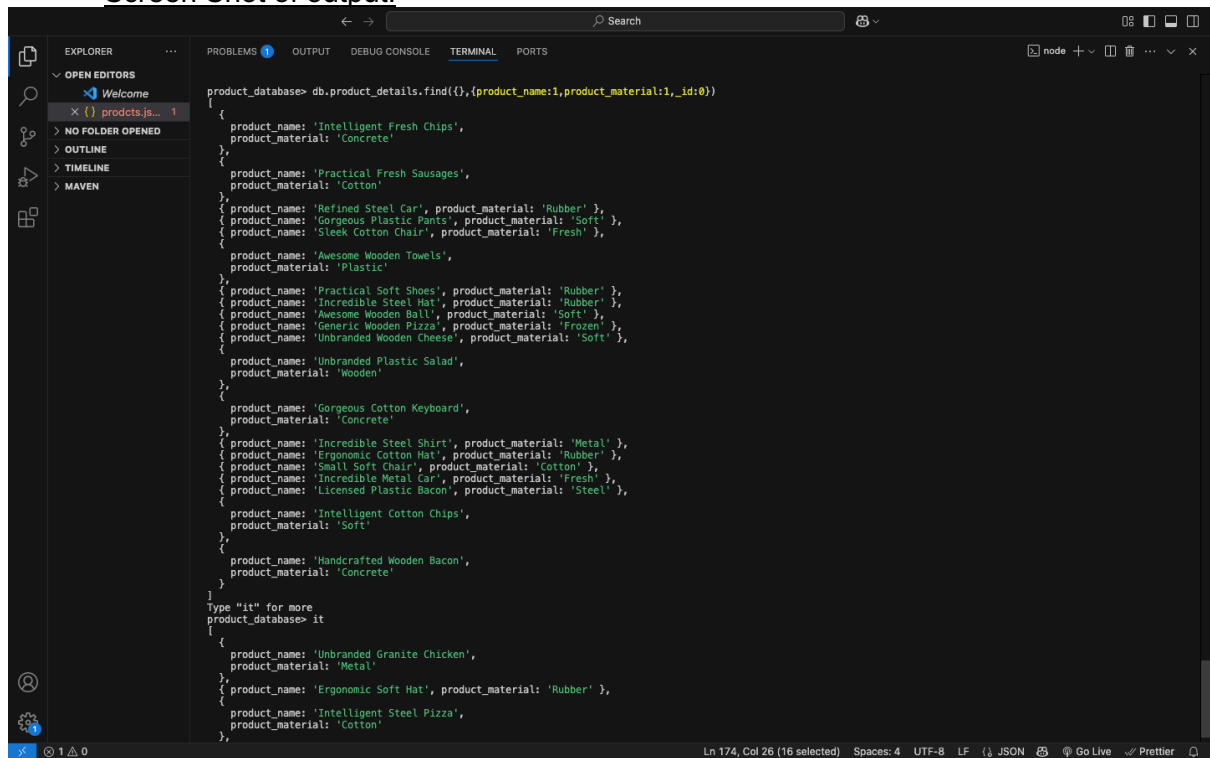
The screenshot shows a VS Code interface with a terminal window. The terminal displays the command `product_database> db.product_details.find({product_price : {$gt:500 }}).limit(4)` and its output, which is a JSON array of four product objects. The products are: 1. 'Intelligent Fresh Chips' (price: 655, material: 'Concrete', color: 'mint green'), 2. 'Practical Fresh Sausages' (price: 911, material: 'Cotton', color: 'indigo'), 3. 'Refined Steel Car' (price: 690, material: 'Rubber', color: 'gold'), and 4. An empty object. The status bar at the bottom indicates the file is `products.js...` and the cursor is at line 177, column 2.

```
product_database> db.product_details.find({product_price : {$gt:500 }}).limit(4)
[
  {
    _id: ObjectId("680f02180cb84fd23337e170"),
    id: '1',
    product_name: 'Intelligent Fresh Chips',
    product_price: 655,
    product_material: 'Concrete',
    product_color: 'mint green'
  },
  {
    _id: ObjectId("680f02f30cb84fd23337e171"),
    id: '2',
    product_name: 'Practical Fresh Sausages',
    product_price: 911,
    product_material: 'Cotton',
    product_color: 'indigo'
  },
  {
    _id: ObjectId("680f02f30cb84fd23337e172"),
    id: '3',
    product_name: 'Refined Steel Car',
    product_price: 690,
    product_material: 'Rubber',
    product_color: 'gold'
  },
  {}
]
product_database>
```

5. Find the product name and product material of each products?

Command: db.product_details.find({}, {product_name:1, product_material:1, id:0})

Screen Shot of output:

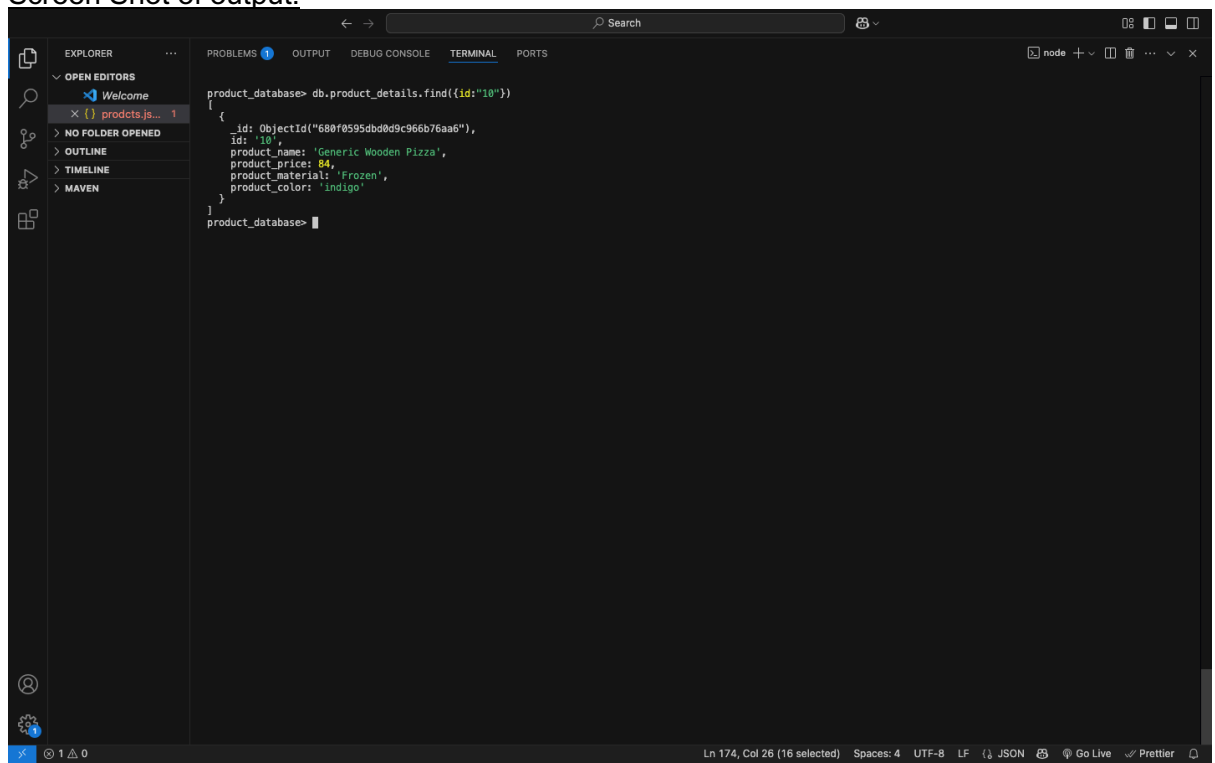


```
product_database> db.product_details.find({}, {product_name:1, product_material:1, id:0})
{
  product_name: 'Intelligent Fresh Chips',
  product_material: 'Concrete'
},
{
  product_name: 'Practical Fresh Sausages',
  product_material: 'Cotton'
},
{
  product_name: 'Refined Steel Car', product_material: 'Rubber' },
{
  product_name: 'Gorgeous Plastic Pants', product_material: 'Soft' },
{
  product_name: 'Sleek Cotton Chair', product_material: 'Fresh' },
{
  product_name: 'Awesome Wooden Towels',
  product_material: 'Plastic'
},
{
  product_name: 'Practical Soft Shoes', product_material: 'Rubber' },
{
  product_name: 'Incredible Steel Hat', product_material: 'Rubber' },
{
  product_name: 'Awesome Wooden Ball', product_material: 'Soft' },
{
  product_name: 'Generic Wooden Pizza', product_material: 'Frozen' },
{
  product_name: 'Unbranded Wooden Cheese', product_material: 'Soft' },
{
  product_name: 'Unbranded Plastic Salad',
  product_material: 'Wooden'
},
{
  product_name: 'Gorgeous Cotton Keyboard',
  product_material: 'Concrete'
},
{
  product_name: 'Incredible Steel Shirt', product_material: 'Metal' },
{
  product_name: 'Ergonomic Cotton Hat', product_material: 'Rubber' },
{
  product_name: 'Small Soft Chair', product_material: 'Cotton' },
{
  product_name: 'Incredible Metal Car', product_material: 'Fresh' },
{
  product_name: 'Licensed Plastic Bacon', product_material: 'Steel' },
{
  product_name: 'Intelligent Cotton Chips',
  product_material: 'Soft'
},
{
  product_name: 'Handcrafted Wooden Bacon',
  product_material: 'Concrete'
}
]
Type "it" for more
product_database> it
{
  product_name: 'Unbranded Granite Chicken',
  product_material: 'Metal'
},
{
  product_name: 'Ergonomic Soft Hat', product_material: 'Rubber' },
{
  product_name: 'Intelligent Steel Pizza',
  product_material: 'Cotton'
},
},
```

6. Find the product with a row id of 10?

Command: db.product_details.find({id:"10"})

Screen Shot of output:



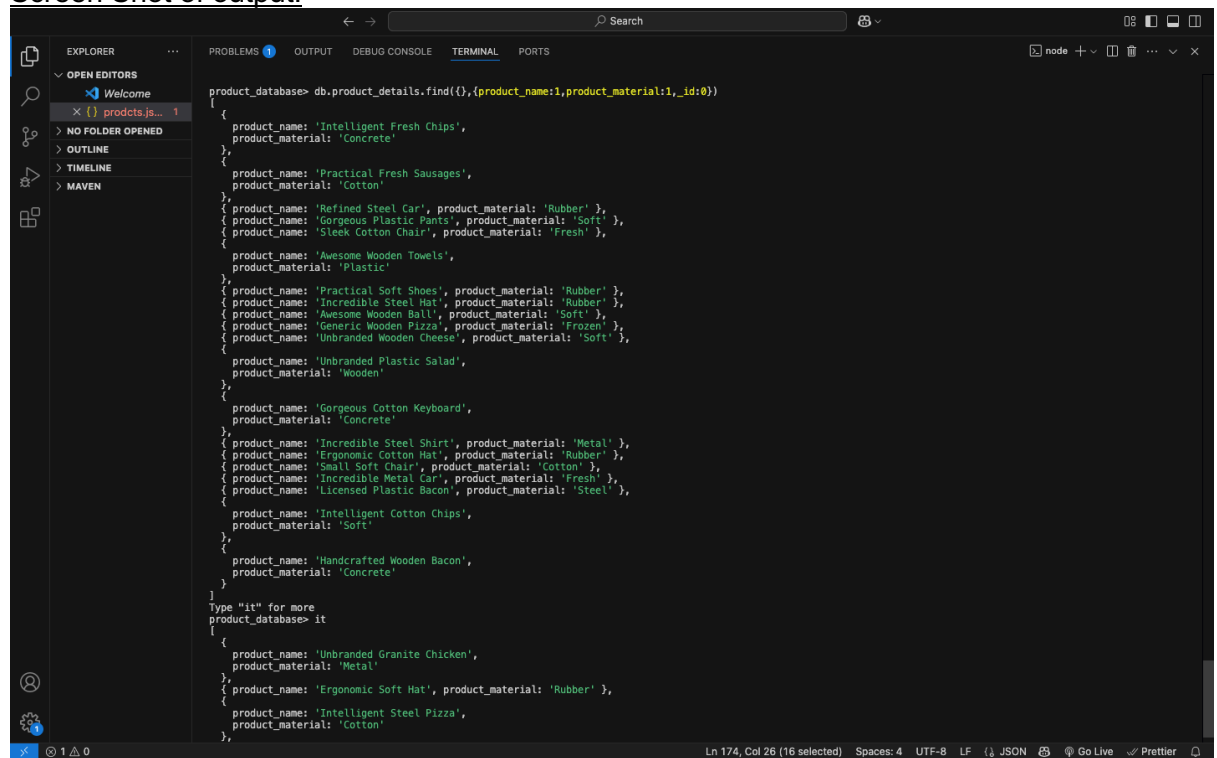
The screenshot shows a VS Code window with a terminal open. The terminal displays the command `product_database> db.product_details.find({id:"10"})` and its output, which is a JSON document representing a product. The document has the following fields: `_id` (ObjectId), `id` (10), `product_name` (Generic Wooden Pizza), `product_price` (84), `product_material` (Frozen), and `product_color` (indigo). The status bar at the bottom indicates the file is at line 174, column 26, with 16 characters selected, using UTF-8 encoding and LF line endings.

```
product_database> db.product_details.find({id:"10"})
{
  _id: ObjectId("680f0595dbd8d9c966b76aa6"),
  id: '10',
  product_name: 'Generic Wooden Pizza',
  product_price: 84,
  product_material: 'Frozen',
  product_color: 'indigo'
}
product_database>
```

7. Find only the product name and product material?

Command: `db.product_details.find({}, {product_name:1, product_material:1, id:0})`

Screen Shot of output:



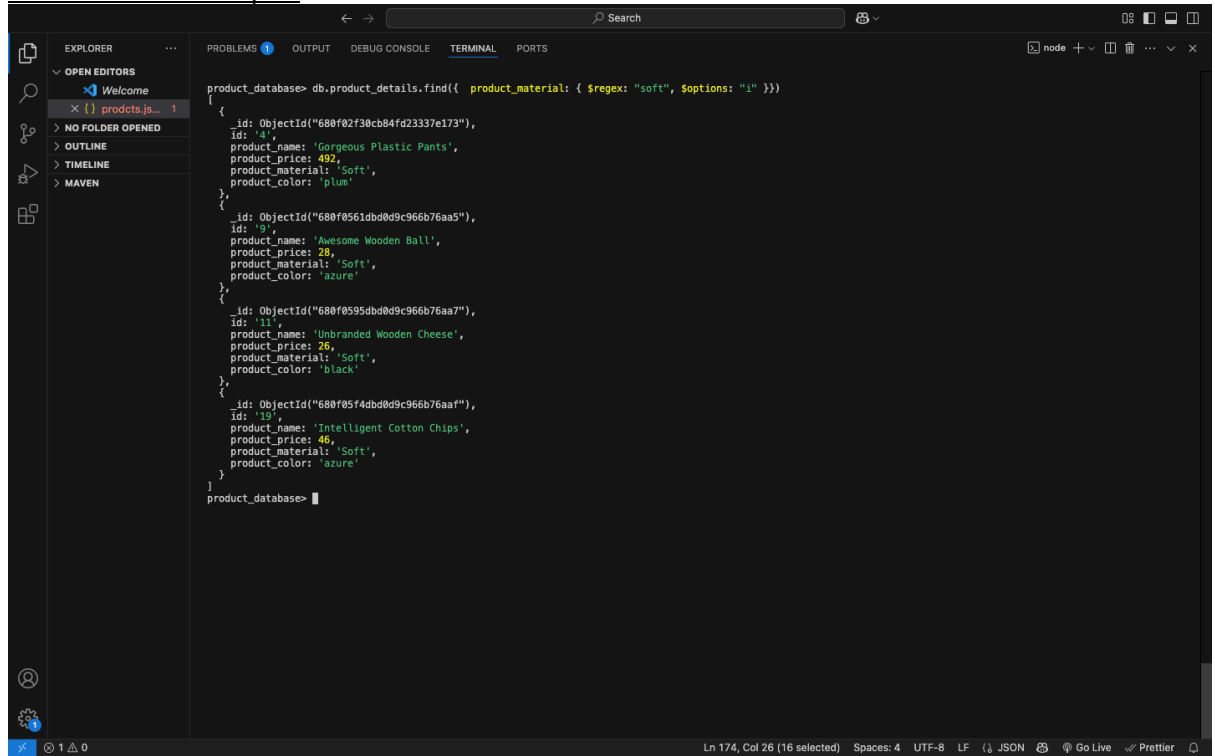
The screenshot shows a Visual Studio Code editor with a terminal window open. The terminal displays the output of a MongoDB query. The query is `product_database> db.product_details.find({}, {product_name:1, product_material:1, id:0})`. The output is a JSON array of objects, each containing `product_name` and `product_material`. The status bar at the bottom indicates the cursor is at line 174, column 26, with 16 characters selected. The editor is configured with 4 spaces, UTF-8 encoding, and the JSON language.

```
product_database> db.product_details.find({}, {product_name:1, product_material:1, id:0})
[
  {
    product_name: 'Intelligent Fresh Chips',
    product_material: 'Concrete'
  },
  {
    product_name: 'Practical Fresh Sausages',
    product_material: 'Cotton'
  },
  {
    product_name: 'Refined Steel Car', product_material: 'Rubber' },
  {
    product_name: 'Gorgeous Plastic Pants', product_material: 'Soft' },
  {
    product_name: 'Sleek Cotton Chair', product_material: 'Fresh' },
  {
    product_name: 'Awesome Wooden Towels',
    product_material: 'Plastic'
  },
  {
    product_name: 'Practical Soft Shoes', product_material: 'Rubber' },
  {
    product_name: 'Incredible Steel Hat', product_material: 'Rubber' },
  {
    product_name: 'Awesome Wooden Ball', product_material: 'Soft' },
  {
    product_name: 'Generic Wooden Pizza', product_material: 'Frozen' },
  {
    product_name: 'Unbranded Wooden Cheese', product_material: 'Soft' },
  {
    product_name: 'Unbranded Plastic Salad',
    product_material: 'Wooden'
  },
  {
    product_name: 'Gorgeous Cotton Keyboard',
    product_material: 'Concrete'
  },
  {
    product_name: 'Incredible Steel Shirt', product_material: 'Metal' },
  {
    product_name: 'Ergonomic Cotton Hat', product_material: 'Rubber' },
  {
    product_name: 'Small Soft Chair', product_material: 'Cotton' },
  {
    product_name: 'Incredible Metal Car', product_material: 'Fresh' },
  {
    product_name: 'Licensed Plastic Bacon', product_material: 'Steel' },
  {
    product_name: 'Intelligent Cotton Chips',
    product_material: 'Soft'
  },
  {
    product_name: 'Handcrafted Wooden Bacon',
    product_material: 'Concrete'
  }
]
Type "it" for more
product_database> it
[
  {
    product_name: 'Unbranded Granite Chicken',
    product_material: 'Metal'
  },
  {
    product_name: 'Ergonomic Soft Hat', product_material: 'Rubber' },
  {
    product_name: 'Intelligent Steel Pizza',
    product_material: 'Cotton'
  },
]
```


8. Find all products which contain the value of soft in product material ?

Command: db.product_details.find({ product_material: { \$regex: "soft", \$options: "i" }})

Screen Shot of output:



The screenshot shows a VS Code interface with a terminal window. The terminal displays the output of a MongoDB query. The query is: `product_database> db.product_details.find({ product_material: { $regex: "soft", $options: "i" }})`. The output is a JSON array of four product objects. Each object contains fields for `_id`, `id`, `product_name`, `product_price`, `product_material`, and `product_color`. The products found are: 'Gorgeous Plastic Pants' (price 492, color plum), 'Awesome Wooden Ball' (price 28, color azure), 'Unbranded Wooden Cheese' (price 26, color black), and 'Intelligent Cotton Chips' (price 46, color azure).

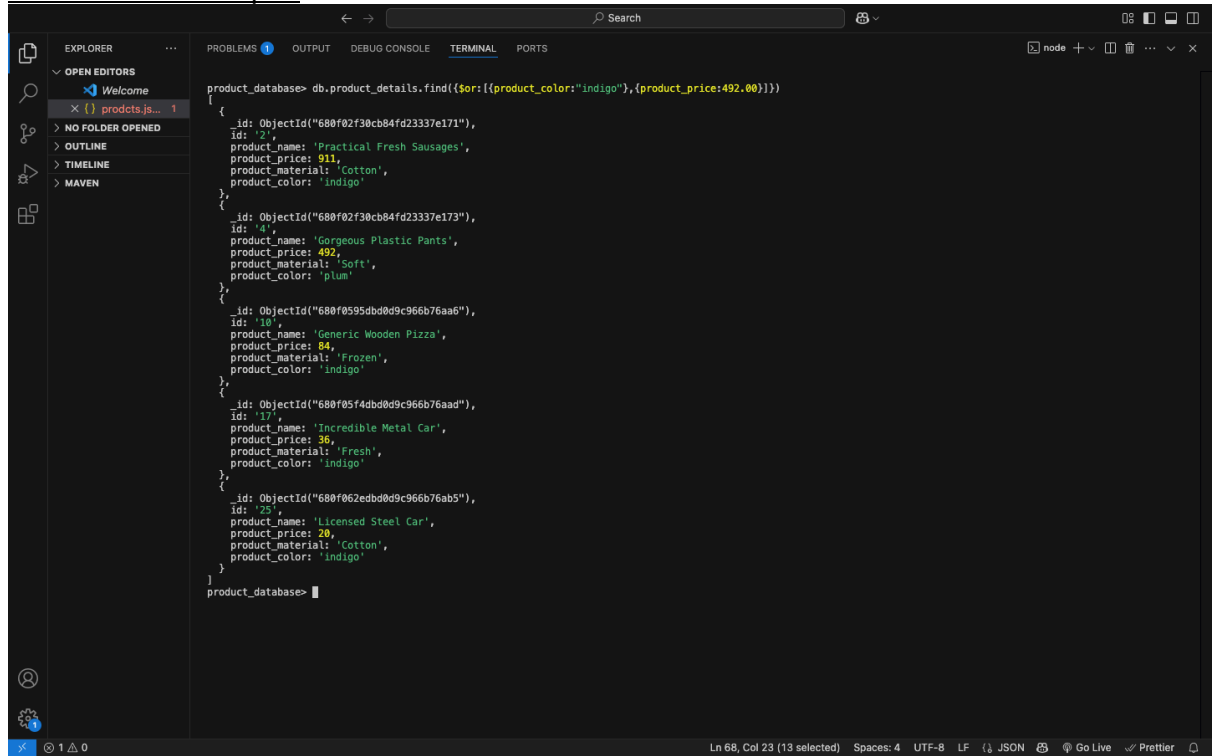
```
product_database> db.product_details.find({ product_material: { $regex: "soft", $options: "i" }})
[
  {
    _id: ObjectId("680f02f30cb84fd23337e173"),
    id: '4',
    product_name: 'Gorgeous Plastic Pants',
    product_price: 492,
    product_material: 'Soft',
    product_color: 'plum'
  },
  {
    _id: ObjectId("680f0561dbd8d9c966b76aa5"),
    id: '9',
    product_name: 'Awesome Wooden Ball',
    product_price: 28,
    product_material: 'Soft',
    product_color: 'azure'
  },
  {
    _id: ObjectId("680f0595dbd8d9c966b76aa7"),
    id: '11',
    product_name: 'Unbranded Wooden Cheese',
    product_price: 26,
    product_material: 'Soft',
    product_color: 'black'
  },
  {
    _id: ObjectId("680f05f4dbd8d9c966b76aa7"),
    id: '19',
    product_name: 'Intelligent Cotton Chips',
    product_price: 46,
    product_material: 'Soft',
    product_color: 'azure'
  }
]
product_database>
```

9. Find products which contain product color indigo and product price 492.00?

Command:

db.product_details.find({\$or:[{product_color:"indigo"},{product_price:492.00}]})

Screen Shot of output:



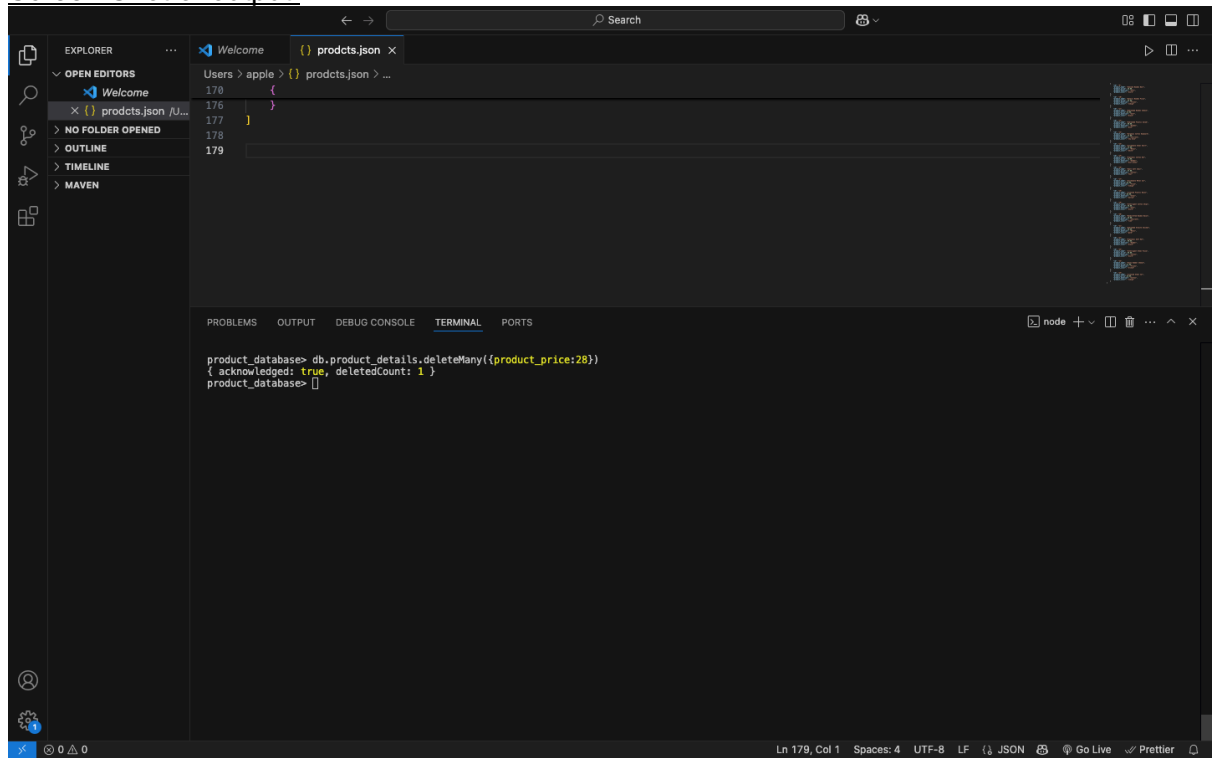
The screenshot shows a Visual Studio Code editor with a terminal window open. The terminal displays the output of a MongoDB query. The query is `product_database> db.product_details.find({$or:[{product_color:"indigo"},{product_price:492.00}]})`. The output is a JSON array containing five product objects. The first object is for 'Practical Fresh Sausages' with price 911 and color 'indigo'. The second is for 'Gorgeous Plastic Pants' with price 492 and color 'plum'. The third is for 'Generic Wooden Pizza' with price 84 and color 'indigo'. The fourth is for 'Incredible Metal Car' with price 36 and color 'indigo'. The fifth is for 'Licensed Steel Car' with price 20 and color 'indigo'. The status bar at the bottom indicates the cursor is at line 68, column 23, with 13 characters selected.

```
product_database> db.product_details.find({$or:[{product_color:"indigo"},{product_price:492.00}]})
[
  {
    _id: ObjectId("680f02f30cb84fd23337e171"),
    id: '2',
    product_name: 'Practical Fresh Sausages',
    product_price: 911,
    product_material: 'Cotton',
    product_color: 'indigo'
  },
  {
    _id: ObjectId("680f02f30cb84fd23337e173"),
    id: '4',
    product_name: 'Gorgeous Plastic Pants',
    product_price: 492,
    product_material: 'Soft',
    product_color: 'plum'
  },
  {
    _id: ObjectId("680f0595dbd0d9c966b76aa6"),
    id: '10',
    product_name: 'Generic Wooden Pizza',
    product_price: 84,
    product_material: 'Frozen',
    product_color: 'indigo'
  },
  {
    _id: ObjectId("680f05f4dbd0d9c966b76aad"),
    id: '17',
    product_name: 'Incredible Metal Car',
    product_price: 36,
    product_material: 'Fresh',
    product_color: 'indigo'
  },
  {
    _id: ObjectId("680f062edbd0d9c966b76ab5"),
    id: '25',
    product_name: 'Licensed Steel Car',
    product_price: 20,
    product_material: 'Cotton',
    product_color: 'indigo'
  }
]
product_database>
```

10. Delete the products which product price value are 28?

Command: db.product_details.deleteMany({product_price:28})

Screen Shot of output:



The screenshot shows the Visual Studio Code interface with a terminal window at the bottom. The terminal displays the output of a MongoDB command. The command executed is `product_database> db.product_details.deleteMany({product_price:28})`. The output shows `{ acknowledged: true, deletedCount: 1 }` and `product_database> []`. The Explorer sidebar on the left shows the project structure with 'products.json' open. The terminal window has tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, and PORTS, with the TERMINAL tab selected. The status bar at the bottom indicates the current line and column (Ln 179, Col 1) and the encoding (UTF-8).

```
product_database> db.product_details.deleteMany({product_price:28})
{ acknowledged: true, deletedCount: 1 }
product_database> []
```