Backend Task:-

1. **Initialize the Database**

```
const axios = require('axios');
const Product = require('./models/Product'); // Product model for MongoDB

app.get('/api/init-db', async (req, res) => {
  try {
    const response = await
axios.get('https://s3.amazonaws.com/roxiler.com/product_transaction.json');
    const transactions = response.data;

    // Seed the database
    await Product.insertMany(transactions);

    res.status(200).json({ message: "Database initialized successfully" });
  } catch (error) {
    res.status(500).json({ message: "Error initializing database", error });
  }
});
```

2. **List All Transactions with Search and Pagination**

```
app.get('/api/transactions', async (req, res) => {
  const { search = '', page = 1, perPage = 10 } = req.query;

  const query = {
    $or: [
      { title: { $regex: search, $options: 'i' } },
      { description: { $regex: search, $options: 'i' } },
      { price: { $regex: search, $options: 'i' } }
    ]
  };

  const transactions = await Product.find(query)
    .skip((page - 1) * perPage)
    .limit(parseInt(perPage));

  res.status(200).json({ transactions });
});
```

3. **API for Monthly Statistics**

```
app.get('/api/stats/:month', async (req, res) => {
  const { month } = req.params;

  const stats = await Product.aggregate([
    {
      $project: {
        month: { $month: "$dateOfSale" },
        price: 1,
        isSold: 1
      }
    },
    { $match: { month: parseInt(month) } },
    {
      $group: {
        _id: null,
        totalSales: { $sum: { $cond: ['$isSold', '$price', 0] } },
        soldItems: { $sum: { $cond: ['$isSold', 1, 0] } },
        notSoldItems: { $sum: { $cond: ['$isSold', 0, 1] } }
      }
    }
  ]);

  res.status(200).json(stats[0]);
});
```

4. **API for Bar Chart**

```
app.get('/api/bar-chart/:month', async (req, res) => {
  const { month } = req.params;

  const barData = await Product.aggregate([
    {
      $project: {
        month: { $month: "$dateOfSale" },
        price: 1
```

```
   }
  },
  { $match: { month: parseInt(month) } },
  {
   $bucket: {
    groupBy: "$price",
    boundaries: [0, 100, 200, 300, 400, 500, 600, 700, 800, 900, Infinity],
    default: "901-above",
    output: { count: { $sum: 1 } }
   }
  }
 ]);

 res.status(200).json(barData);
});
```

5. **API for Pie Chart**

```
app.get('/api/pie-chart/:month', async (req, res) => {
 const { month } = req.params;

 const pieData = await Product.aggregate([
  {
   $project: {
    month: { $month: "$dateOfSale" },
    category: 1
   }
  },
  { $match: { month: parseInt(month) } },
  {
   $group: {
    _id: "$category",
    count: { $sum: 1 }
   }
  }
 ]);

 res.status(200).json(pieData);
});
```

6. **API for Combined Data**

```
app.get('/api/combined/:month', async (req, res) => {
  const { month } = req.params;

  const [stats, barChart, pieChart] = await Promise.all([
    axios.get(`/api/stats/${month}`),
    axios.get(`/api/bar-chart/${month}`),
    axios.get(`/api/pie-chart/${month}`)
  ]);

  res.status(200).json({
    statistics: stats.data,
    barChart: barChart.data,
    pieChart: pieChart.data
  });
});
```

Database Schema Design (Example):

```
const mongoose = require('mongoose');

const productSchema = new mongoose.Schema({
  productID: String,
  title: String,
  description: String,
  category: String,
  price: Number,
  dateOfSale: Date,
  isSold: Boolean
});

const Product = mongoose.model('Product', productSchema);

module.exports = Product;
```

Front-End Task:-

1 Setting Up React

```
npx create-react-app transaction-dashboard
cd transaction-dashboard
npm install axios chart.js react-chartjs-2
```

2. App Structure

Basic structure of the front-end page, including:

- Transactions Table
- Transaction Statistics
- Bar Chart (for price ranges)
- Pie Chart (for categories)

3. Code Implementation

```
 import React, { useState, useEffect } from 'react';

 import axios from 'axios';

 import TransactionsTable from './TransactionsTable';

import TransactionsStatistics from './TransactionsStatistics';

import BarChart from './BarChart';

import PieChart from './PieChart';



const App = () => {

  const [month, setMonth] = useState('March'); // Default to March

  const [searchTerm, setSearchTerm] = useState('');

  const [transactions, setTransactions] = useState([]);

  const [statistics, setStatistics] = useState({});

  const [barChartData, setBarChartData] = useState({});
  const [pieChartData, setPieChartData] = useState({});
 const months = ['January', 'February', 'March', 'April', 'May', 'June', 'July', 'August', 'September', 'October', 'November', 'December'];
  useEffect(() => {
    fetchTransactions();
    fetchStatistics();
    fetchBarChart();
    fetchPieChart();
  }, [month, searchTerm]);
  const fetchTransactions = async () => {
    try {
      const response = await axios.get('/api/transactions', {
        params: { month, search: searchTerm }
      });
      setTransactions(response.data);
    } catch (error) {
      console.error('Error fetching transactions:', error);
    }
  };
const fetchStatistics = async () => {
    try {
      const response = await axios.get(`/api/statistics?month=${month}`);
      setStatistics(response.data);
    } catch (error) {
      console.error('Error fetching statistics:', error);
    }
  };
 const fetchBarChart = async () => {
    try {
      const response = await axios.get(`/api/bar-chart?month=${month}`);
      setBarChartData(response.data);
    } catch (error) {
      console.error('Error fetching bar chart data:', error);
    }
  };
 const fetchPieChart = async () => {
    try {
      const response = await axios.get(`/api/pie-chart?month=${month}`);
      setPieChartData(response.data);
    } catch (error) {
      console.error('Error fetching pie chart data:', error);
    }
  };
return (
    <div>
      <h1>Transactions Dashboard</h1>
   {/* Month Selector */}
```

```
      <select value={month} onChange={(e) => setMonth(e.target.value)}>
        {months.map((m) => (
          <option key={m} value={m}>{m}</option>
        ))}
      </select>
  {/* Transactions Table */}
      <TransactionsTable transactions={transactions} searchTerm={searchTerm} setSearchTerm={setSearchTerm} />
  {/* Statistics */}
      <TransactionsStatistics statistics={statistics} />
 {/* Bar Chart */}
      <BarChart data={barChartData} />
  {/* Pie Chart */}
      <PieChart data={pieChartData} />
    </div>
  );
};
export default App;
```

4. TransactionsTable Component

```
import React from 'react';
const TransactionsTable = ({ transactions, searchTerm, setSearchTerm }) => {
  return (
    <div>
      <h2>Transactions Table</h2>
```

```
     {/* Search Input */}
     <input
       type="text"
       placeholder="Search transactions"
       value={searchTerm}
       onChange={(e) => setSearchTerm(e.target.value)}
     />
{/* Table Display */}
     <table>
       <thead>
         <tr>
           <th>Title</th>
           <th>Description</th>
           <th>Price</th>
         </tr>
       </thead>
       <tbody>
         {transactions.map((transaction) => (
           <tr key={transaction.id}>
             <td>{transaction.title}</td>
             <td>{transaction.description}</td>
             <td>{transaction.price}</td>
```

```
      </tr>
     ))}
    </tbody>
   </table>
  </div>
 );
};
export default TransactionsTable;
```

5. TransactionsStatistics Component

```
import React from 'react';
const TransactionsStatistics = ({ statistics }) => {
 return (
  <div>
   <h2>Transactions Statistics</h2>
   <div>Total Sale Amount: {statistics.totalSales}</div>
   <div>Total Sold Items: {statistics.soldItems}</div>
   <div>Total Not Sold Items: {statistics.notSoldItems}</div>
  </div>
 );
};
export default TransactionsStatistics;
```

6. BarChart Component

```
import React from 'react';
import { Bar } from 'react-chartjs-2';


const BarChart = ({ data }) => {
  const chartData = {
    labels: ['0-100', '101-200', '201-300', '301-400', '401-500', '501-600', '601-700', '701-800', '801-900', '901-above'],
    datasets: [
      {
        label: 'Number of Items',
        data: data.itemsInRange,
        backgroundColor: 'rgba(75, 192, 192, 0.6)',
      },
    ],
  };
return <Bar data={chartData} />;
};
export default BarChart;
```

7. PieChart Component

```
import React from 'react';
import { Pie } from 'react-chartjs-2';
const PieChart = ({ data }) => {
  const chartData = {
    labels: data.categories.map(cat => cat.name),
    datasets: [
      {
        label: 'Items per Category',
        data: data.categories.map(cat => cat.items),
        backgroundColor: ['#FF6384', '#36A2EB', '#FFCE56', '#4BC0C0', '#9966FF'],
      },
    ],
  };
  return <Pie data={chartData} />;
};
export default PieChart;
```