TruEstate – SDE Intern Assignment

Role: Software Development Engineer Intern (6 Months + Performance-Based PPO)
Submission Deadline: 08 December 2025, 11:59 PM (IST)

Assignment Overview

Your task is to develop a **Retail Sales Management System** that demonstrates essential software engineering capabilities across both frontend and backend components.
The system must support advanced **Search**, **Filtering**, **Sorting**, and **Pagination** functionalities based on the dataset specifications provided.

Your implementation should reflect:

- Strong foundational problem-solving and engineering approach
- Clean, maintainable, and modular architecture
- UI built according to provided structural guidelines
- Accurate and efficient handling of structured sales data
- Professional execution aligned with real-world SDE responsibilities

This assignment represents challenges similar to those faced in production-grade systems at TruEstate.

Dataset, Interface & Design

Dataset Link

https://drive.google.com/file/d/1tzbyuxBmrBwMSXbL22r33FUMtO0V_lxb/view?usp=sharing

Sales Data Requirements

Use the provided dataset, which includes the following attributes:

Customer Fields:

- Customer ID
- Customer Name
- Phone Number
- Gender
- Age
- Customer Region
- Customer Type

Product Fields:

- Product ID
- Product Name
- Brand
- Product Category
- Tags

Sales Fields:

- Quantity
- Price per Unit
- Discount Percentage
- Total Amount
- Final Amount

Operational Fields:

- Date
- Payment Method
- Order Status
- Delivery Type
- Store ID
- Store Location
- Salesperson ID
- Employee Name

Your system must correctly interpret and process all these attributes.

Functional Requirements

1. Search

Implement full-text search across the following fields:

- Customer Name
- Phone Number

Search must be:

- Case-insensitive
- Accurate
- Performant

- Able to work alongside filters and sorting

2. Filters (Multi-Select)

Implement multi-select or range-based filtering for:

- Customer Region
- Gender
- Age Range
- Product Category
- Tags
- Payment Method
- Date Range

Filters must:

- Work independently
- Work in combination
- Maintain state alongside sorting and search

3. Sorting

Implement sorting for:

- Date (Newest First)
- Quantity
- Customer Name (A–Z)

Sorting must preserve active search and filters.

4. Pagination

- Page size: **10 items per page**
- Must support Next / Previous navigation
- Must retain active search, filter, and sort states

UI Requirements

Figma Link

https://www.figma.com/design/6MSR9Lhy8tc5gdOepEyy6q/Assignment?node-id=0-1&p=f&t=ISiFSTN2Ol2BSPjN-0

Follow a clear, minimal, and structured layout:

- Search Bar

- Filter Panel
- Transaction Table (List/Grid)
- Sorting Dropdown
- Pagination Controls

The UI may be styled creatively, but must align with the provided structure.

Engineering Requirements

Your solution must demonstrate professional coding standards:

General

- Clear separation of frontend and backend responsibilities
- Clean, readable, maintainable code
- Predictable state management
- No duplicate logic for filtering or sorting
- Avoid unnecessary nesting or complexity
- Use best coding practices

Project Structure (Single Repository)

Your submission must follow this exact structure:

```
□ root/
├── backend/
│   ├── src/
│   │   ├── controllers/
│   │   ├── services/
│   │   ├── utils/
│   │   ├── routes/
│   │   ├── models/       (if required)
│   │   └── index.js      (entry file)
│   │
│   ├── package.json
│   └── README.md
│
├── frontend/
│   ├── src/
│   │   ├── components/
│   │   ├── pages/        (if Next.js)
│   │   ├── routes/       (if React Router)
│   │   ├── services/
│   │   ├── utils/
│   │   ├── hooks/
│   │   ├── styles/
│   │   └── main.jsx / App.jsx
│   │
│   ├── public/
│   ├── package.json
│   └── README.md
│
```

```
│
├── docs/
│   └── architecture.md
│
├── README.md
└── package.json (if monorepo setup)
```

Edge Case Expectations

Your implementation must correctly handle:

- No search results
- Conflicting filters
- Invalid numeric ranges
- Large filter combinations
- Missing optional fields

Submission Structure

1. Live Application URL

A deployed, working version of the full application.

2. GitHub Repository URL

A public repository containing the full project.

README.md — Required Format

Your README must contain **only**:

```
1. Overview (3–5 lines)
2. Tech Stack
3. Search Implementation Summary
4. Filter Implementation Summary
5. Sorting Implementation Summary
6. Pagination Implementation Summary
7. Setup Instructions
```

4. Architecture Document

Located at: /docs/architecture.md
Must contain:

- Backend architecture
- Frontend architecture
- Data flow
- Folder structure
- Module responsibilities

Important Notes

- Auto-generated tools or one-click app builders are **not allowed**.
- All logic must be developed by the candidate.
- Near-identical submissions will be rejected.

All the best

This assignment is crafted to evaluate your clarity of thought, coding discipline, and execution quality — essential attributes for an SDE Intern contributing to scalable systems at TruEstate.