**Part 1: Introduction + Solution Design**

**Hi everyone!  
Thank you so much for joining.**

**Today, I am going to walk you through my Salesforce project called:**

**“HandsMen Threads: Elevating the Art of Sophistication in Men’s Fashion.”**

This project is all about building a custom Salesforce solution for a fictional high-end men's fashion brand — where we handle everything from customer management, product inventory, orders, and even loyalty campaigns.

Throughout the demo, I will show you how I designed the data model, added automation, used flows and Apex, and made sure everything is secure using roles and permission sets.

We will go step-by-step, just like a real-world Salesforce implementation.

So, let us start with the **Solution Design Document**, where I planned the whole system before building it.

**📄 Solution Design Document (Spoken Style)**

Alright, so here is my **Solution Design Document**.

I broke it down into sections to match a real Salesforce project.

Let us go through each part quickly.

**🟨 1. Business Goal**

HandsMen Threads wanted a Salesforce solution to:

* Track customer orders.
* Manage their product inventory in real time.
* Automatically send emails when orders are confirmed or stock is low.
* And run loyalty or marketing campaigns using automation.

So, I built the solution using custom objects, automation, flows, email alerts, and a batch job.

**🟨 2. Objects Used**

Here are the custom objects I created:

1. **HandsMen Customer** – to store customer info like name, email, phone, and loyalty status.
2. **HandsMen Product** – to manage their product catalog.
3. **HandsMen Order** – to track what customers have ordered.
4. **Inventory** – to track how much stock is available.
5. **Marketing Campaign** – to plan future loyalty programs.

I also created **tabs** for each object so users can access them easily.

**🟨 3. Relationships & Fields**

* I used **lookup relationships** to link Orders with Customers and Products.
* I added **formula fields** to calculate total quantity or order value.
* There are also fields like **loyalty status**, **email**, **phone**, and more.

So, each object is fully connected — just like in a real business system.

**🟨 4. Security**

To make sure the right people have access to the right data:

* I created **Roles** for Sales, Inventory, and Marketing teams.
* I assigned **Profiles** and **Permission Sets** to give proper object-level permissions.
* I added **2 sample users** – Niklaus and Kol – and assigned them the Platform user profile with permission sets.

So, the system is secure and permission-based.

**🟨 5. Automation**

Then, I added automation for 4 user stories.

We will go through each one later, but here is a quick list:

1. When an order is confirmed, send an email to the customer.
2. When stock is low, send an alert to the inventory manager.
3. Automatically restock inventory using a scheduled **Batch Apex** job.
4. Run loyalty programs or marketing campaigns using **flows**.

So, I used flows, email alerts, and Apex code to automate these scenarios.

**🟨 6. App Setup**

I created a custom **Lightning App** called **HandsMen Threads**, where all users can access the:

* Objects
* Tabs
* Reports
* Flows

Everything is nicely organized for business users.

✅ So that is the full solution design — the plan behind the system.

Now I will show you how I implemented each part step-by-step.

Let us begin with the first **user story**: sending a confirmation email when an order is confirmed.