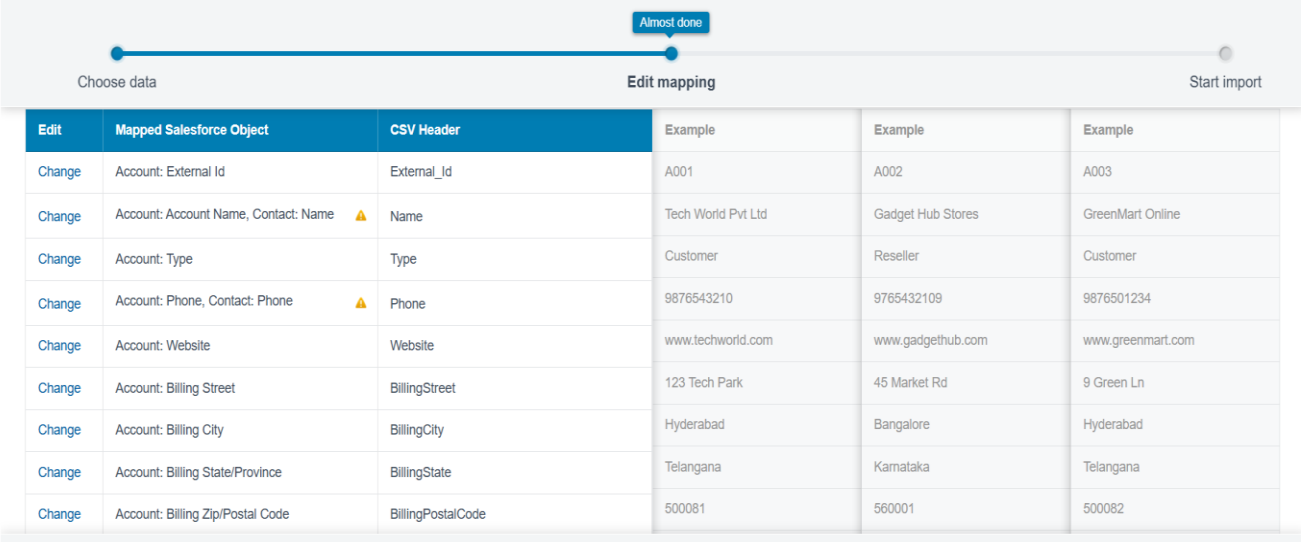
Phase 8 and 9: Data Management , Deployment & Reporting, Dashboards ,Security Review

Phase 8: Data Management & Deployment

This phase covered the management of project data and the deployment of metadata changes. Our strategy focused on modern development practices to ensure version control and a robust deployment pipeline.

8.1 Data Import

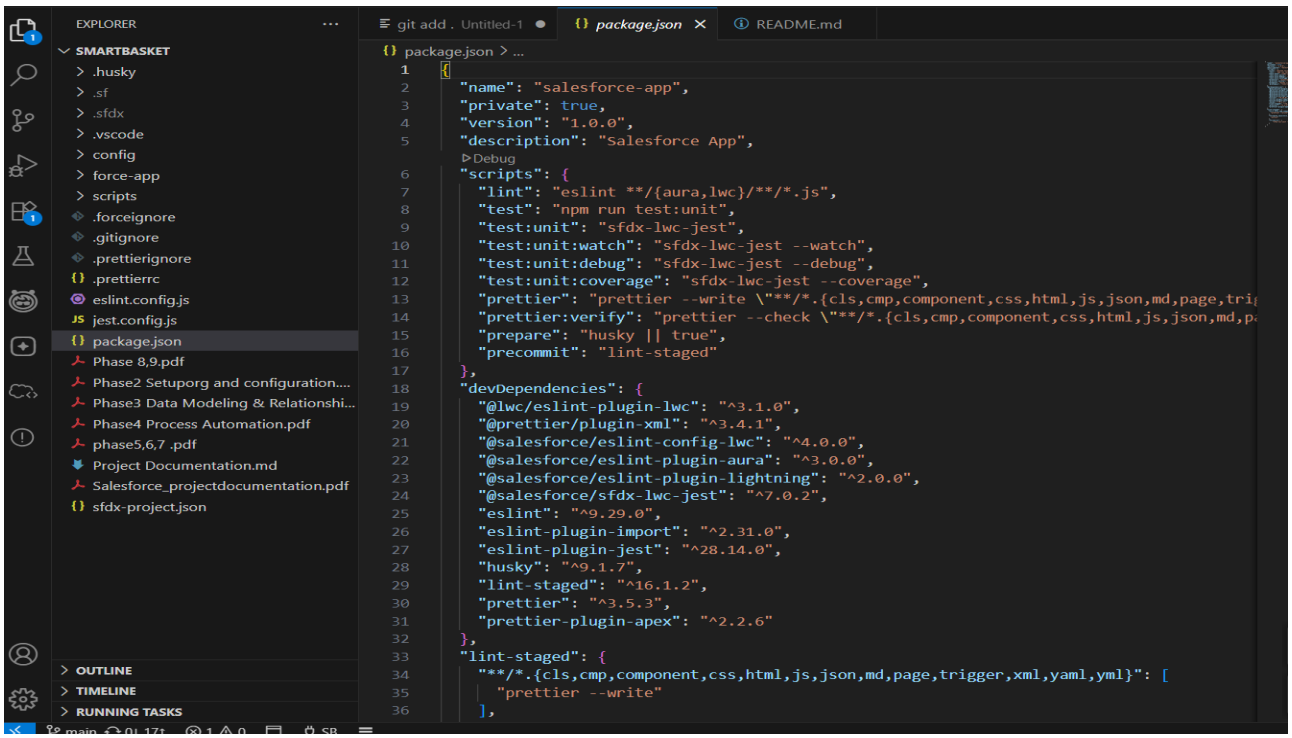
- **Data Import Wizard** was used to load Accounts, Contacts, and custom Inventory Balance records.



Edit	Mapped Salesforce Object	CSV Header	Example	Example	Example
Change	Account: External Id	External_Id	A001	A002	A003
Change	Account: Account Name, Contact: Name	Name	Tech World Pvt Ltd	Gadget Hub Stores	GreenMart Online
Change	Account: Type	Type	Customer	Reseller	Customer
Change	Account: Phone, Contact: Phone	Phone	9876543210	9765432109	9876501234
Change	Account: Website	Website	www.techworld.com	www.gadgethub.com	www.greenmart.com
Change	Account: Billing Street	BillingStreet	123 Tech Park	45 Market Rd	9 Green Ln
Change	Account: Billing City	BillingCity	Hyderabad	Bangalore	Hyderabad
Change	Account: Billing State/Province	BillingState	Telangana	Karnataka	Telangana
Change	Account: Billing Zip/Postal Code	BillingPostalCode	500081	560001	500082

- **Data Loader** was used for objects not supported by the Wizard, such as Products, Orders, and Order Items.
- Sample datasets were successfully uploaded, making the application demo-ready with real records.
- **Deployment**: Instead of traditional methods like Change Sets, we used a source-driven deployment model.
    - **Salesforce CLI (SFDX)**: We used SFDX commands to retrieve and deploy metadata, which provided a more granular and controlled approach to development.
    - **VS Code**: The project was developed within Visual Studio Code, a modern IDE with robust support for Salesforce development.
    - **Git & GitHub**: We leveraged **Git** for version control, tracking all metadata changes locally. The project's repository on **GitHub** served as the single

source of truth for all metadata, enabling collaborative development and a reliable backup.



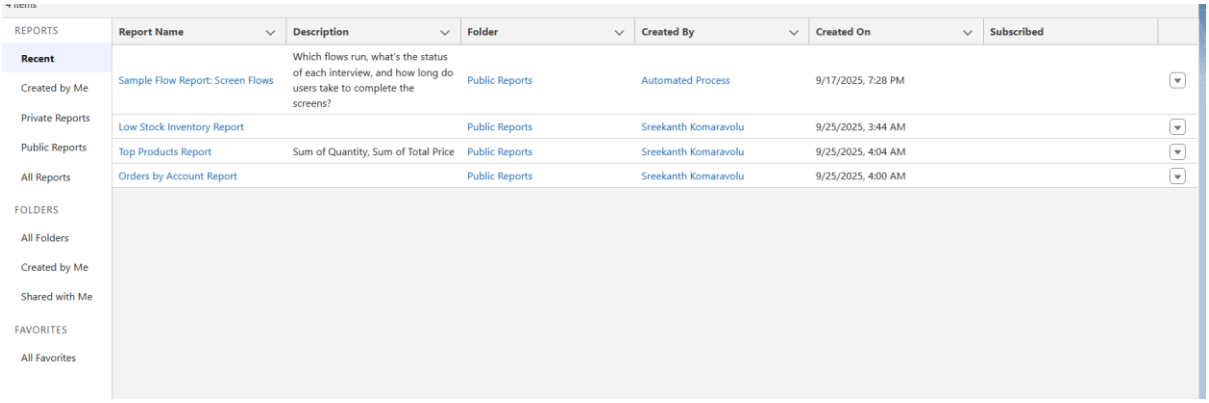
- **Ant Migration Tool**: We did not use the Ant Migration Tool. Our deployment strategy was based entirely on the Salesforce CLI (SFDX) and Git, which is the recommended and more modern approach for Salesforce development. The Ant Migration Tool is an older method that has largely been superseded by the flexibility of SFDX.

Phase 9: Reporting, Dashboards & Security Review

This phase focused on creating reports and dashboards to monitor SmartBasket operations.

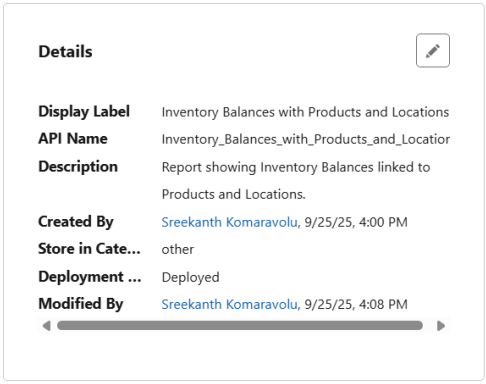
1. **Reports**

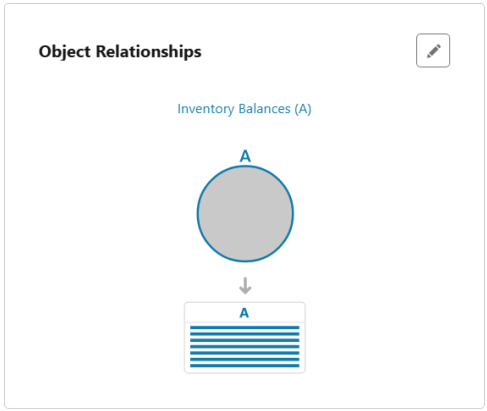
- Low Stock Inventory Report (Tabular) → highlights products with stock below reorder level.
- Orders by Account (Tabular/Summary) → groups orders by customer.
- Top Products Report (Summary) → aggregates product sales quantities and revenue.



2. Report Types

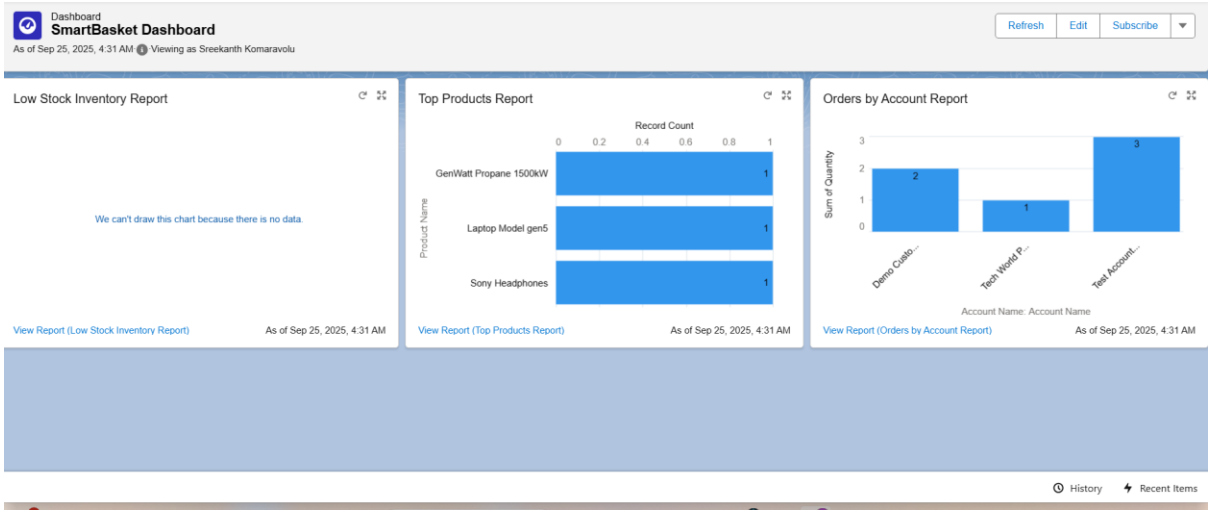
- Inventory Balance with Product & Location → custom report type built to combine stock with product & warehouse details.
- Orders with Order Products → custom report type for order header and line item reporting.





3. Dashboards

- SmartBasket Dashboard created with:
  - Low Stock Inventory (Donut).
  - Orders by Account (Bar Chart).
  - Top Products (Bar Chart).



4. Security Review

We configured the security settings to ensure that the right users have the right level of access to data and functionality.

- Profiles & Permission Sets: We used Profiles for base access to objects and tabs and Permission Sets for granular permissions. For example, the System Administrator profile has broad "View All" access to objects, while specific users may have more limited access.
- Organization-Wide Defaults (OWD): The OWD for the Order object was set to Public Read Only to allow all internal users to see all orders, while record access was managed through a combination of ownership and sharing settings.
- Sharing Settings: We can use Sharing Rules to grant specific groups of users access to records they don't own. This was considered for granting the System Administrator full access to all records if the OWD was set to private, but the Public Read Only setting addressed this.
- Field Level Security: Field-level security was used to control which fields users can see and edit on an object. For example, a procurement user may have read-only access to some fields, while a manager has edit access.
- Session Settings: We configured the session settings to ensure secure access, including setting login IP ranges to restrict user access from untrusted locations.