Project Report: Laptop Request Catalog Item

Project Title: ServiceNow Laptop Request Automation

Team Members

- JAYAKIRUBA V (Team Leader)
- SRINITHI K S
- SRINITHI R
- KAVIPRIYA P

Phase 1: Ideation Phase

Objective:
To automate the laptop request process within the organization using ServiceNow, replacing the existing manual, time-consuming procedure.

Problem Identification:
- Manual and inefficient process.
- No dynamic fields or real-time data validation.
- High chances of errors and inconsistent records.
- Lack of governance or change tracking.

Proposed Solution:
To create a Service Catalog Item in ServiceNow that allows employees to request laptops easily. The form should be dynamic based on user input, include UI Policies for field visibility, provide a reset option for better user experience, and support Update Sets for deployment and governance.

Phase 2: Project Planning Phase

Goals:
- Develop a reusable and automated catalog item.
- Reduce manual intervention in IT service requests.
- Improve data accuracy and form efficiency.

Resources Required:
- ServiceNow Development Instance
- Access to Service Catalog, UI Policies, UI Actions
- Update Set for tracking configurations

Milestones:
1. Requirement understanding and documentation.
2. Service Catalog Item creation.
3. Dynamic form behavior implementation.
4. Testing and validation.
5. Final deployment.

Risk Assessment:
- Data validation issues → addressed through UI Policies.
- Version conflicts during migration → resolved using Update Sets.

Phase 3: Project Design Phase

System Design Components:

1. **Catalog Item Design:**
   - Name: Laptop Request
   - Category: Hardware
   - Description: Use this item to request a new laptop.

2. **Dynamic Form Logic:**
   - Implemented via Catalog UI Policy
   - Condition: When 'Additional Accessories' = true
   - Action: Show and make mandatory the field 'Accessories Details'.

3. **User Experience Enhancement:**
   - Added Reset Form button using UI Action.
   - JavaScript:

```
function resetForm() {
  g_form.clearForm();
  alert('The form has been reset.');
}
```

4. **Governance:**
   - Used Update Set named 'Laptop Request' to track all configurations.

Phase 4: Requirement Analysis Phase

Functional Requirements:

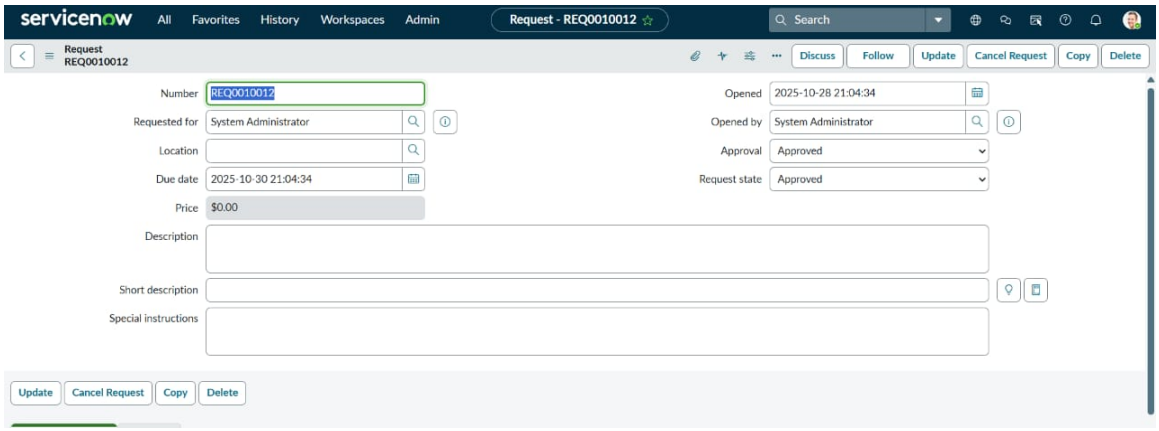
- Dynamic Form: Form should show/hide fields based on user input.
- Form Reset: User can reset the form easily.
- Tracking: Changes should be captured using Update Sets.
- Validation: Mandatory fields must be enforced dynamically.
- Deployment: Configuration should be transferable between instances.

Non-Functional Requirements:

- Usability: Simple and intuitive user interface.
- Reliability: Ensure correct data submission.
- Maintainability: Easy to update and reuse in other requests.

Implementation Steps:

1. Create and activate the Update Set.

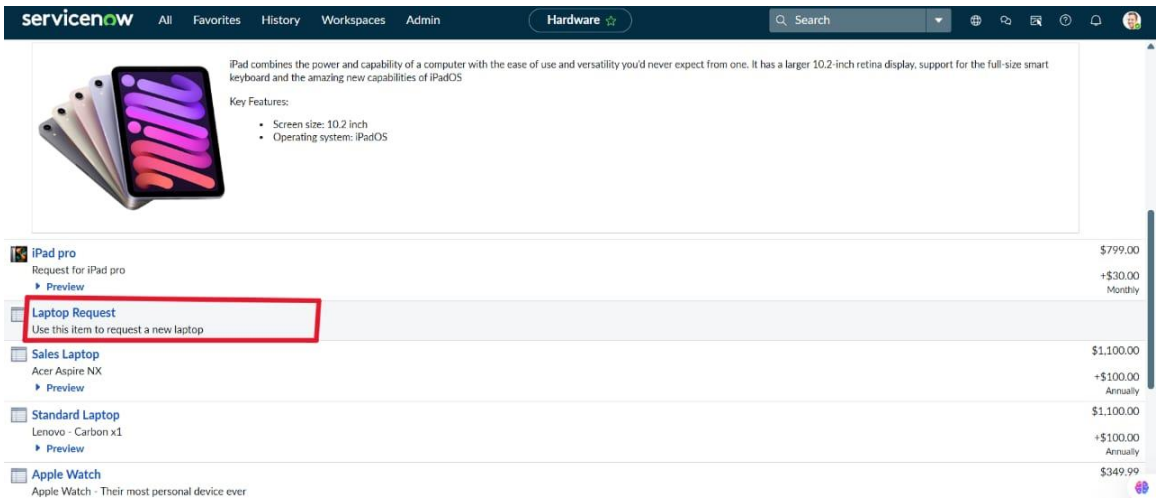


2. Configure the Catalog Item under Hardware.

3. Add variables and UI Policies.

4. Implement UI Action for form reset.

5. Test, validate, and deploy via Update Set.



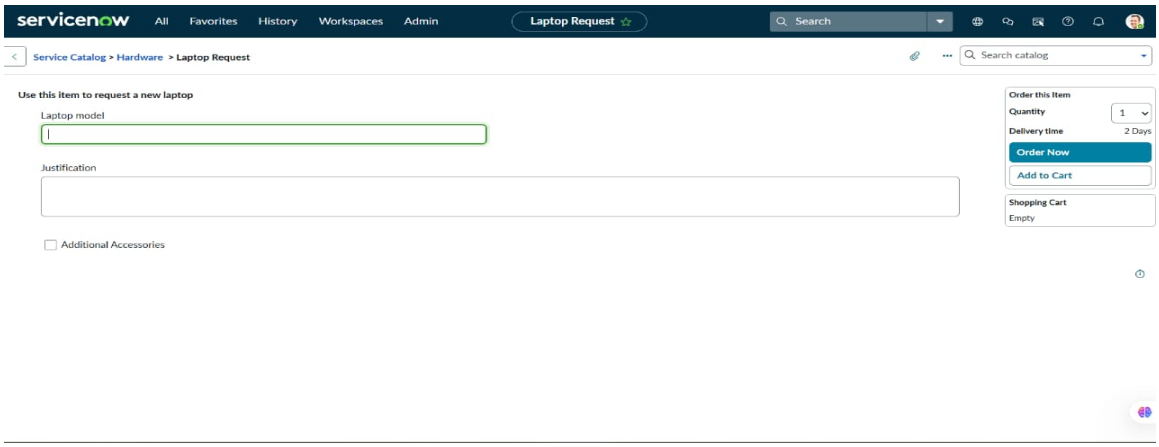
Phase 5: Performance Testing Phase

Testing Objectives:

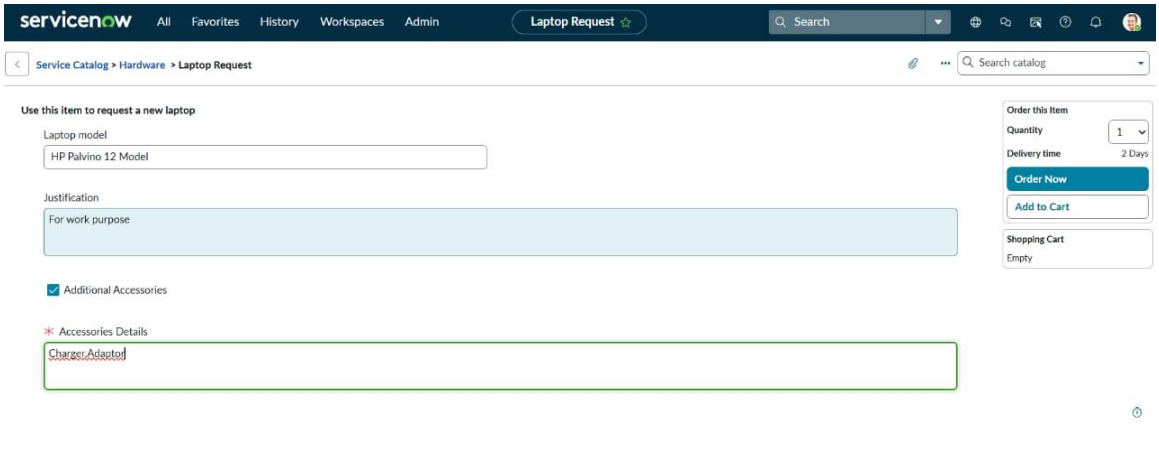
To ensure that the Service Catalog Item functions correctly and efficiently after deployment.

Testing Scenarios:

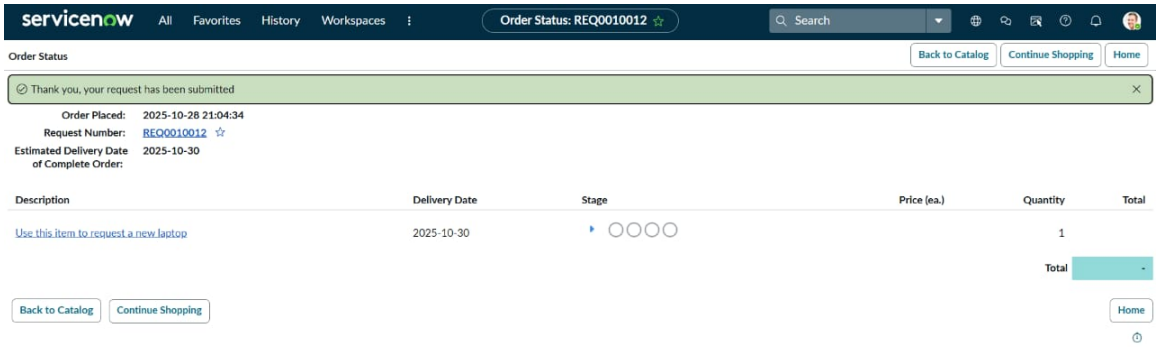
- Form loads successfully → All fields visible as per design.



- Additional Accessories unchecked → Accessories Details hidden.
- Additional Accessories checked → Accessories Details visible and mandatory.



- Reset button clicked → Form cleared successfully.
- Form submitted → Request recorded properly.



Outcome:

- All functionalities worked as expected.
- No performance lag during form load or submission.
- Dynamic UI and reset functionality improved user experience.

Result

Manual effort reduced significantly. Improved data accuracy through form automation. Easy migration and governance ensured using Update Sets. Enhanced overall user experience.