

Project Title: Laptop Request Catalog Item

Team ID:NM2025TMID07314

Team Members: 4

Team Leader: Shabu SH

Team Member 1: Jason J

Team Member 2: Pravin F

Team Member 3:Berlin JO BJ

Problem Statement: Using a laptop catalog to enquire about laptops and accessories

Objective: To develop a small ServiceNow app for checking new laptops and accessories

Skills: ServiceNow basics, JavaScript, reporting.

TASK INITIATION

Quick assumptions (so instructions are consistent)

You have admin access to a ServiceNow dev instance (or the instance SmartInternz provided).

You will use Service Catalog + Flow Designer + Asset/CMDB for fulfilment.

Keep names consistent (use u_ prefix for custom fields/variables if your org requires it).

1 — Plan the item (5 – 10 min)

1. Objective: Allow users to request a laptop with required approvals and automated fulfilment task creation.

2. Fields (variables) to collect:

Requester (auto-filled) — requester (use current user)

Department — department (choice or reference)

Job Role / Justification — justification (mandatory for high-end)

Laptop Category — laptop_category (choices: Standard, High-end, Developer)

Preferred Model — preferred_model (single-line text or choice)

RAM — ram_size (choice: 8GB/16GB/32GB)

Storage — storage_size (choice)

Required By Date — required_date (date/time)

Cost Center / Budget Code — budget_code

Manager Approval Required? (implied by flow)

3. Workflow overview: Request → Manager approval → Procurement approval

(if cost > threshold) → Create fulfilment task → Create asset & assign → Notify requester.

4. Acceptance criteria: End-to-end request creates tasks and asset record; requester can track status.

2 — Create the Catalog Item (Service Catalog)

1. Navigate: Service Catalog > Catalog Definitions > Maintain Items → New.

2. Set:

Name: Laptop Request

Catalogs: Hardware (or appropriate)

Category: IT Hardware

Short description: “Request a laptop for work/study”

Active: true

3. Save.

3 — Add Variables (form fields)

1. From the item record, go to Variables related list → New for each variable.

Use these recommended settings:

Display name	Type	Name (example)	Notes
Requester	Lookup (user)	requester	Default value = javascript:gs.getUserID() or use current.requested_for
Department	Lookup or Choice	department	Optional reference to Dept table
Laptop Category	Choice	laptop_category	Choices: Standard, High-end, Developer
Preferred Model	Single Line	preferred_model	Optional
RAM	Choice	ram_size	8GB, 16GB, 32GB
Storage	Choice	storage_size	256GB, 512GB, 1TB
Required By Date/Time		required_date	validation: not in past
Justification	Multi-Line	justification	required conditionally
Budget Code	Single Line	budget_code	optional

2. Mark required fields appropriately (e.g., requester, laptop_category).

4 — UX polish: Catalog Item form layout & help

1. Arrange variables into sections (e.g., Requester Info, Laptop Specs, Business Info).
2. Add Question Help text/tooltips for ambiguous fields (e.g., “Provide model only if you have a preference”).
3. Add an image/icon for the item to improve UI.

5 — Add UI Policies & Data Validation

1. UI Policy: If laptop_category == "High-end" → make justification mandatory & visible.

Create: UI Policies → New → Condition: laptop_category is High-end → Action: Set mandatory for justification.

2. Client Script (on Catalog Item) to validate required_date not in the past:

```
function onSubmit() {  
  
    var reqDate = g_form.getValue('required_date');  
  
    if (!reqDate) return true;  
}
```

```
var now = new Date();

var r = new Date(reqDate);

if (r < now) {

    alert('Required By date cannot be in the past.');

    return false;

}

return true;

}
```

3. Client Script to auto-populate requester if needed (useful if not defaulted).

6 — Create Approval Flow (Flow Designer)

Use Flow Designer (recommended) not old Workflow Editor.

1. Navigate: Flow Designer > New → Laptop Request - Approval & Fulfilment.
2. Trigger: When a sc_request (or sc_req_item) is created for this Catalog Item (filter by catalog item sys_id).
3. Steps (example sequence):

Get variables (optional step to map values)

Approval - Manager: Use Approvals action: Approval - Manager or a custom approval record.

If manager approves → Check cost condition (if budget_code implies cost or laptop_category == 'High-end') → Procurement Approval (if needed).

On final approval: Create Task: sc_task or custom u_fulfilment_task. Populate task fields: short_description, assigned_to (Procurement/Asset Team), due_date.

Create/Update Asset (if you automate asset creation): use Create Record action on alm_asset or cmdb class, map model, serial (if available). Or create a manual Asset Provision task.

Notifications: Send notification to requester and assignees (use email template or in-system notification).

Update request item: set state to Approved/Rejected/Fulfilled at appropriate points.

4. Use subflows for repeated logic: e.g., a reusable Approval Chain subflow used across items.

1. Create Email Notifications: When Request Item state changes to Approved, Rejected, Fulfilled. Include: item number, status, approver name, expected delivery date, link to request.
2. Also create a notifier to the assignee when a fulfilment task is assigned.

8 — Fulfilment tasks & Asset provisioning

1. When approval completes, Flow creates an sc_task assigned to Hardware Fulfilment group. Fields: u_requested_model, u_ram_size, u_storage, u_required_by.
2. Fulfilment Agent actions: procure laptop, tag asset, update asset record,

change task to Closed Complete.

3. Optionally automate asset creation: if you have an integration or predefined asset pool, the flow can set Asset to assigned and mark request fulfilled.

9 — Security & Roles

1. Use ACL to ensure only appropriate roles see sensitive variables (budget_code maybe restricted to finance roles).

2. Restrict who can approve: manager role or group (use approver fields mapped to user manager or department head).

10 — Testing (end-to-end)

Create a test plan and execute these tests:

1. Unit tests:

Form validations (required fields, date not past).

UI Policy (justification required for High-end).

2. Approval tests:

Submit request as a user → approval request is sent to manager.

Approve and check next approval; reject and ensure requester notified.

3. Fulfilment test:

Final approval creates fulfilment task assigned to correct group.

When task closed, request item moves to Fulfilled and requester gets notification.

4. Negative tests:

Missing required fields; wrong date; non approver tries to approve.

5. Edge cases:

High-end request without justification → must block.

Large requisition triggers procurement approval.

Record results and screenshots.

11 — Logging & Troubleshooting

1. Add `gs.log()` (server scripts) or `console.log()` (client) during development for debugging. Remove or minimize logs later.
2. Check Flow Designer executions and logs for failures. Use `sys_flow_context`

to view flow runs.

12 — Documentation & Screenshots

Create a README or project doc that includes:

1. Project overview & objective.
2. Variables list (table with variable name, type, mandatory).
3. Flow diagram (simple image showing approvals & tasks).

4. Key scripts (client & server) and where they are used.

5. Test cases and results (with screenshots).

6. Installation or import instructions (if you export update sets or provide JSON).

7. GitHub link (if you publish).

GitHub repo structure suggestion

SN-LaptopRequestCatalogItem/

 └── README.md

 └── docs/

 | └── flow-diagram.png

 | └── screenshots/

```
|   └── test-results.md  
  
├── config/  
|   ├── variables.json (list of variables)  
  
|   └── flow-mapping.md  
  
└── scripts/  
    ├── client-scripts/  
    |   └── required_date_validate.js  
    └── server-scripts/  
        └── create_asset_snippet.js  
  
└── export/  
    └── update_set.xml (if you export from ServiceNow)
```

13 — Sample server-side snippet (create asset)

> Use carefully — adapt field names to your instance.

```
// Script Action or Business Rule (server side)

(function execute(inputs, outputs) {

    var asset = new GlideRecord('alm_asset');

    asset.initialize();

    asset.name = inputs.preferred_model || 'Laptop';

    asset.model_id = inputs.model_sysid; // optional

    asset.assignment_group = 'Hardware Fulfilment';

    asset.assigned_to = inputs.requester_sysid;

    asset.u_ram = inputs.ram_size;

    asset.u_storage = inputs.storage_size;

    asset.insert();

})(inputs, outputs);
```

14 — Submission checklist (what to deliver)

Finalized Catalog Item in ServiceNow (item link or sys_id).

Screenshots: form, UI policy effect, approval notification, fulfilled request, created asset.

Flow Designer screenshots (flow canvas + sample run).

README.md with steps to reproduce + assumptions.

Optional: exported Update Set or JSON that contains your item/flow.

GitHub repo link (if included) with above structure and test results.

15 — Future enhancements (short list)

Auto-assign asset from inventory pool instead of creating new.

Integration with procurement system for purchase orders.

SLA timers and escalation for approval delays.

Dashboard for managers to view pending laptop requests.

Add analytics (counts by department, monthly spend).

16 — Estimated times (guideline)

Planning & variables: 30 – 60 min

Catalog item + variables + UI policies: 60 – 90 min

Flow Designer approvals + notifications: 60 – 120 min

Fulfilment tasks & asset script: 60 min

Testing & documentation: 90 – 120 min