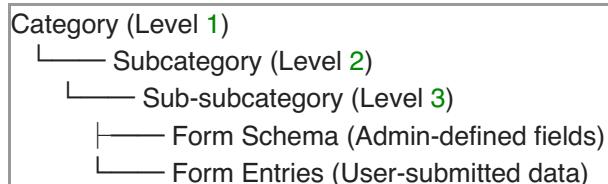


CA Services - 3-Level Hierarchy Update

Overview

The CA Services section has been updated to support a **3-level hierarchical structure** with dynamic form schemas and form entries.

Structure



Hierarchy Levels

Level 1: Categories

- **GST** - GST Registration, Filing, and Compliance Services
- **Income Tax** - Income Tax Return Filing and PAN Services
- **Company Registration** - Company and Business Registration Services

Level 2: Subcategories

GST:

- GST Registration
- GST Filing
- GST Cancellation

Income Tax:

- ITR Filing
- PAN Services

Company Registration:

- Private Limited Company
- One Person Company
- Partnership Firm

Level 3: Sub-subcategories

GST Registration:

- Proprietorship
- Partnership Firm
- Private Limited Company
- One Person Company

GST Filing:

- Monthly Filing
- Quarterly Filing

GST Cancellation:

- Normal Cancellation
- Surrender Registration

ITR Filing:

- ITR-1
- ITR-2
- ITR-3

PAN Services:

- New PAN
- PAN Correction

Form Schema

Each **Sub-subcategory** has a **Form Schema** that defines:

- Field types: text, email, phone, number, date, select, textarea, file, checkbox
- Field validation rules
- Required fields
- Field sections/groups
- Help text and placeholders

Example Form Schema Structure:

```
{  
  "subSubcategoryId": "sub_subcategory_id",  
  "fields": [  
    {  
      "name": "businessName",  
      "label": "Business Name",  
      "type": "text",  
      "isRequired": true,  
      "validation": { "minLength": 3, "maxLength": 100 },  
      "section": "business_details",  
      "order": 1  
    }  
  ],  
  "sections": [  
    {  
      "id": "business_details",  
      "title": "Business Details",  
      "order": 1  
    }  
  ]  
}
```

Form Entries

Users can submit **Form Entries** based on the Form Schema:

- **Draft:** Save incomplete forms
- **Submitted:** Submit completed forms for processing
- **Status:** draft, submitted, in_review, approved, rejected

New API Endpoints

Catalog APIs

- GET /api/v1/ca/category - Get all categories (Level 1)
- GET /api/v1/ca/category/:categoryId/subcategory - Get subcategories (Level 2)
- GET /api/v1/ca/subcategory/:subcategoryId/sub-subcategory - Get sub-subcategories (Level 3)

Form Schema APIs

- GET /api/v1/ca/forms/schema/:subSubcategoryId - Get form schema for a sub-subcategory
- POST /api/v1/ca/forms/schema - Create/update form schema (Admin only)

Form Entry APIs

- GET /api/v1/ca/forms/entries - Get user's form entries
- GET /api/v1/ca/forms/entries/:entryId - Get form entry details
- POST /api/v1/ca/forms/entries - Submit form entry
- POST /api/v1/ca/forms/entries/draft - Save draft entry

Models

CAServiceCategory

- Added level field: 'category' | 'subcategory' | 'sub_subcategory'
- Supports 3-level hierarchy via parentCategoryId

CAFormSchema

- Stores admin-defined form fields
- One schema per sub-subcategory
- Supports versioning

CAFormEntry

- Stores user-submitted form data
- Links to sub-subcategory and form schema
- Tracks submission status

Seed Data

Run the seed script to populate test data:

```
node seed-ca-data.js
```

This creates:

- 3 Categories
- 8 Subcategories
- 13 Sub-subcategories
- 2 Form Schemas (GST Registration - Proprietorship, ITR-1)
- 2 Form Entries (1 submitted, 1 draft)

Testing

Run the test script:

```
./test-ca-apis.sh
```

Files Changed

New Files

- src/models/CAFormSchema.ts - Form schema model
- src/models/CAFormEntry.ts - Form entry model
- src/controllers/ca/forms.controller.ts - Form management controller
- seed-ca-data.js - Seed data script
- test-ca-apis.sh - Test script

Updated Files

- src/models/CAServiceCategory.ts - Added level field
- src/controllers/ca/catalog.controller.ts - Added getSubSubcategories function
- src/routes/ca.routes.ts - Added form and sub-subcategory routes
- src/validations/ca.validation.ts - Added form validation schemas

Benefits

1. **Flexible Structure:** Easy to add new categories, subcategories, and sub-subcategories
2. **Dynamic Forms:** Admin can define custom forms for each sub-subcategory
3. **Version Control:** Form schemas support versioning for updates
4. **Draft Support:** Users can save incomplete forms and submit later
5. **Scalable:** Supports unlimited hierarchy levels and form fields