**Milestone 2: Public Policy Navigation Using AI**

**Streamlit App – Excel to JSON Converter (Policies)**

**Purpose**

This application enables users to upload Excel files containing structured policy data and convert them into a standardized JSON format. It supports schema extraction, auto-generates missing Policy\_IDs, and offers interactive previews and downloads. The tool is designed for:

* Public policy dashboards
* Machine learning preprocessing
* Educational and government data dissemination

**Functional Modules**

1. Page Setup

* Title clearly communicates the app’s purpose
* Wide layout optimizes visibility for large tables and JSON previews

2. File Upload

* Accepts .xlsx files via drag-and-drop or manual selection
* Displays a prompt if no file is uploaded
* Prevents unsupported formats from entering the pipeline

3. Schema Extraction

* Automatically reads the uploaded Excel file
* Extracts column headers to define the dataset schema
* Displays schema for user confirmation and teaching clarity

4. Record Structuring

* Each row is treated as a standalone policy record
* A unique identifier (Policy\_ID) is enforced across all records
* Missing or empty IDs are auto-generated using a sequential format (e.g., POL001, POL002)

5. Table Preview Options

* Checkbox toggles full dataset view
* Slider allows partial preview of selected number of rows
* Enables quick inspection without overwhelming the interface

6. JSON Preview

* Displays the entire JSON structure in a readable format
* Helps users verify the output before download
* Useful for debugging, validation, and teaching

7. Download Functionality

* Converts structured data into a downloadable JSON string
* File named policies.json
* Ready for use in APIs, dashboards, or ML pipelines

**Output Format Example**

json

{

"policies": [

{

"Policy\_ID": "POL001",

"Scheme\_Name": "XYZ Scheme",

"Target\_Group": "Students",

"Subsidy\_Amount": "₹5,000",

"DBT\_Enabled": "Yes",

...

}

]

}

**Notes**

* Gracefully handles missing Policy\_ID
* Compatible with multilingual and numeric data
* Suitable for public policy, education, and dashboard integration
* Output is structured under a top-level "policies" key for easy parsing

**Approach**

Design Philosophy

The app is built with a modular, schema-aware mindset to support scalable policy data transformation. It emphasizes clarity, error-proofing, and user empowerment—making it ideal for public dashboards, educational tools, and ML preprocessing.

Stepwise Workflow

1. User-Centric Interface
   * Clean layout and guided upload process
   * Feedback prompts for missing files
2. Schema Awareness
   * Column headers extracted and displayed
   * Ensures transparency and validation
3. Record Structuring
   * Each row becomes a policy record
   * Unique Policy\_ID enforced across all entries
4. Preview Logic
   * Full or partial table view based on user selection
   * Slider enables dynamic row control
5. JSON Transparency
   * Full JSON preview before download
   * Ensures user confidence in output
6. Download Enablement
   * One-click export of structured JSON
   * Ready for integration into dashboards or ML pipelines

**UX Design Highlights**

|  |  |  |
| --- | --- | --- |
| Feature | Description | Benefit |
| Wide Layout | Optimized for large tables and JSON previews | Improves readability and navigation |
| Schema Display | Shows extracted column headers | Confirms dataset structure |
| Row Slider | Allows partial preview of data | Speeds up inspection of large files |
| JSON Preview | Displays full structured output | Ensures transparency before download |
| Download Button | One-click export of JSON file | Ready for integration or analysis |

**Integration Potential**

|  |  |
| --- | --- |
| Use Case | Integration Benefit |
| Policy Dashboards | JSON output feeds directly into frontend components |
| ML Preprocessing | Structured records simplify feature engineering |
| Public Sector Portals | Enables transparent scheme dissemination |
| Teaching & Training | Schema logic and preview tools aid instruction |

**Suggested Extensions**

* Sector-wise filters (e.g., Health, Education, Agriculture)
* Toggle for DBT-enabled schemes only
* OCR integration for scanned policy PDFs
* Export options to CSV or XML for legacy systems
* API endpoint integration for real-time data sync