# IOSCO DATA SCRAPING DOCUMENTATION

## 1. Goal

Extract and store all IOSCO i-Scan PDF documents, including extracted text, entity names, and identifiers such as PAN and CIN, into a structured MongoDB database.

## 2. Process Breakdown

### Step 1: Identify Source Page and Extract PDF Links

Used BeautifulSoup and requests to scrape the IOSCO i-Scan page: https://www.iosco.org/i-scan/  
Filtered all anchor tags ending with '.pdf' to build a complete list of document links.

### Step 2: Download PDF Files

All extracted PDF links were downloaded to a local folder using requests and saved with appropriate filenames.

### Step 3: Extract Text from PDFs

Used PyMuPDF (fitz) to extract raw text from each PDF. The text from each page was concatenated and stored as a single string per file.

### Step 4: Extract PAN, CIN, and Entities

Applied regular expressions to extract PAN and CIN codes from the text. Also used the HuggingFace transformer model 'dslim/bert-base-NER' to extract named entities (organizations).

### Step 5: Structure Each PDF’s Data

For each PDF, a dictionary was created with keys: filename, PANs, CINs, organization names, and raw text.

### Step 6: Upload to MongoDB Atlas

Used PyMongo to connect to MongoDB Atlas and inserted all structured records into a collection named 'pdf\_notices' inside the 'iosco\_data' database.

### Step 7: Optional Cleanup

Performed post-processing to flatten single-item lists (e.g., organizations) into strings where applicable for better readability in the database.

## 3. MongoDB Access Code

from pymongo import MongoClient  
  
# Connect to MongoDB Atlas (guest access)  
client = MongoClient("mongodb+srv://guest:guest\_pass@intellewings.enj2zjz.mongodb.net/?retryWrites=true&w=majority&appName=Intellewings")  
  
# Select the database and collection  
db = client["iosco\_data"]  
collection = db["notices"]

# Fetch and print all documents  
docs = collection.find()  
for doc in docs:  
 print(doc)