# IR Search Engine

**Name:** Krishna Sharma
**Reg No:** AP22110010128
**Section:** R

A compact search engine for AI/ML websites demonstrating core Information Retrieval pipelines: **crawling → indexing → ranking → query interface**. Implements **TF-IDF, PageRank, and HITS.**

## Features

| Component | Description |
|---|---|
| Web Crawler | BFS crawling, robots.txt respect, link graph |
| Text Processing | Tokenization, stopword removal, stemming |
| Inverted Index | Term + document frequency mapping |
| Ranking Algorithms | TF-IDF, PageRank, HITS |
| Search Interface | CLI search with 3 ranking modes |

## IR Algorithms Overview

### TF-IDF

- Calculates **term relevance**
- Formula: $\text{TF} \times \log\left(\frac{N}{DF}\right)$

### PageRank

- Link-based **global authority**
- Power-iteration until convergence

### HITS

- Computes **authority + hub** scores on query-relevant subgraph
- **Topic-specific authority**

### Ranking Modes

| Mode | Focus |
|---|---|
| TF-IDF | Text relevance |
| TF-IDF + PageRank | Relevance + global authority |
| TF-IDF + HITS | Relevance + topic-authority |

## Run Instructions

```
python src/crawler.py    # Crawl and store pages
python src/indexer.py    # Build inverted index
python src/pagerank.py   # Compute PageRank
python src/search.py     # Run search
```

## Execution Pipeline

1. Crawl AI/ML pages → build link graph
2. Preprocess text (tokenize, stopwords, stem)
3. Construct inverted index
4. Score with TF-IDF + optional link algorithms
5. Rank & return snippets

## Performance (50 Pages)

| Task | Time |
|---|---|
| Crawling | 2–5 min |
| Indexing | < 30s |
| PageRank | ∼10–20 iterations |
| HITS | ∼5–15 iterations/query |
| Search | < 2s |

## Project Structure

```
src/
  crawler.py
  indexer.py
  ranker.py
  pagerank.py
  hits.py
  search.py
```

## Objective

This project demonstrates real IR system concepts: **Web crawling, inverted indexing, TF-IDF scoring, PageRank, HITS, query processing, and result ranking.**