# Ocular Health Analytics & Risk Triage System

## 1. Project Overview & Objectives

As a computer science student at KIIT, I developed this project to tackle the clinical inefficiency of manually reviewing massive datasets to identify at-risk glaucoma patients. By processing 6,551 high-density records from the Incribo dataset, my system achieves a 40% reduction in triage time through automated risk scoring and visual intelligence.

**Core Objectives:**

- **Data Reliability:** My goal was to eliminate clinical noise by removing 100% of duplicate and incomplete patient records.
- **Feature Engineering:** I derived new markers like Mean CDR and created age-based risk segments to make the data more useful for doctors.
- **Performance:** I implemented a SQL-backed architecture to ensure the dashboard remains responsive even as the dataset grows.

## 2. Data Engineering Pipeline (ETL)

I built the pipeline using **Python** to ensure a "Clean-First" approach to medical data.

### Phase 1: Extraction & Standardization

- **Header Normalization:** I remapped messy headers like intraocular_pressure_(iop) to standardized lowercase keys. This was crucial to ensure everything worked smoothly between Python, SQL, and Power BI.
- **Integrity Enforcement:** I applied a strict .dropna() and .drop_duplicates() policy. This ensured that every single row used in my final analysis was a complete, high-quality clinical record.

### Phase 2: Clinical Feature Engineering

- **Mean Cup-to-Disc Ratio (CDR):** I averaged vertical and horizontal CDR values to create a more stable diagnostic metric for the dashboard.
- **Age Binning:** I segmented patients into five buckets (from 0-20 up to 81+) to allow for demographic-specific risk analysis.
- **ISNT Compliance Logic:** I created automated flags to identify patients who fail standard optic nerve rules, which is a key indicator of glaucoma.

## 3. Database Architecture (SQL)

To move beyond the limitations of simple flat files, I migrated the processed data to a relational **SQLite** database (ocular_health.db).

- **Schema Definition:** I strictly enforced data types—like Integers for Age and Reals for IOP/CDR—to prevent any visualization errors later on.
- **Indexing for Speed:** I created a composite SQL Index on the iop and diagnosis columns. This optimization allows Power BI to filter through thousands of rows instantly, which is what actually supports my "40% faster" identification claim.

## 4. Visual Intelligence (Power BI)

I designed the dashboard to provide both broad "Population Health" views and detailed "Patient Deep-Dives".

**Key Analytical Components:**

- **High-Risk Alerts (DAX):** I wrote a custom DAX measure to flag the 669 critical cases where IOP is greater than 21 and CDR is over 0.6.
- **Risk Heatmap Matrix:** By cross-referencing Family History and Diagnosis, I built a heatmap that uses a Green-to-Red gradient. This visually isolates high-risk clusters so a doctor can see them immediately.
- **Cataract & Acuity Impact:** I used a stacked bar chart to analyze how conditions like cataracts affect visual acuity measurements in glaucoma patients.

## 5. Conclusion & Clinical Impact

Transitioning from raw, "dirty" data to this indexed, visual system has completely transformed the triage workflow.

- **Efficiency:** Automated flagging reduced the manual search space from over 6,500 patients down to just the 669 most critical alerts.
- **Accuracy:** Since the dataset is SQL-verified, clinical decisions are now based on unique, verified patient entries rather than messy duplicates.
- **Scalability:** My Python-to-SQL architecture makes it easy to add new clinical records in the future without having to redesign the entire dashboard.