# Placement Eligibility Stream-lit Application

## Project Overview:

This project is a data-driven **Streamlit web application** designed to evaluate students' placement eligibility based on customizable criteria. It integrates synthetic student performance data and displays insights through interactive dashboards, enabling real-time decision-making for EdTech placement teams.

## Objectives:

* Design and implement a Streamlit application where users can input **eligibility criteria** for placements (e.g., minimum problem-solving scores or soft skills scores).
* Based on these inputs, the app filters and displays the eligible students from a dataset stored in a relational database.

## Technology:

1. Frontend: Stream-lit
2. Backend: Python, SQLite
3. Libraries: Faker, pandas, Random, NumPy
4. Database: SQLite
5. Development Tools: Jupyter Notebook

## Approach:

## Step 1: Dataset Creation

## Generate four interlinked tables using **Faker**:

## 1. **Students Table**: Name, age, gender, city, contact, enrollment and graduation year.

## 2. **Programming Table**: Language, problems solved, assessments, mini projects, certifications, project scores.

## 3. **Soft Skills Table**: Scores for communication, teamwork, presentation, leadership, etc.

## 4. **Placements Table**: Placement status, internship count, interview rounds, company, package, date.

## Step 2: Data Storage

## - Store the data in SQLite

## - Use **Python** for database operations and relationships.

## Step 3: Streamlit Application

## - Accept dynamic input criteria from users (e.g., `problems\_solved > 50`, `communication > 75`)

## - Show a filtered list of \*\*eligible students\*\*.

## - Add download option for filtered results.

## Step 4: SQL Queries & Insights

## Write and display **10 insightful queries**, such as:

1. Students with High Problem Solving Skills (>100 problems solved)

2. Top 10 Students by Placement Package

3. Average Soft Skill Score by Batch

4. Students Without Any Certifications

5. Count of Students per Programming Language

6. Students with Strong Presentation Skills (>80)

7. Internship to Placement Conversion Ratio

8. Students with All-Rounder Skills (High in Programming, Soft Skills, and Mock Interview)

9. Students Without Placement Offers

10. Top 5 Batches by Average Mock Interview Score

## System Architecture:

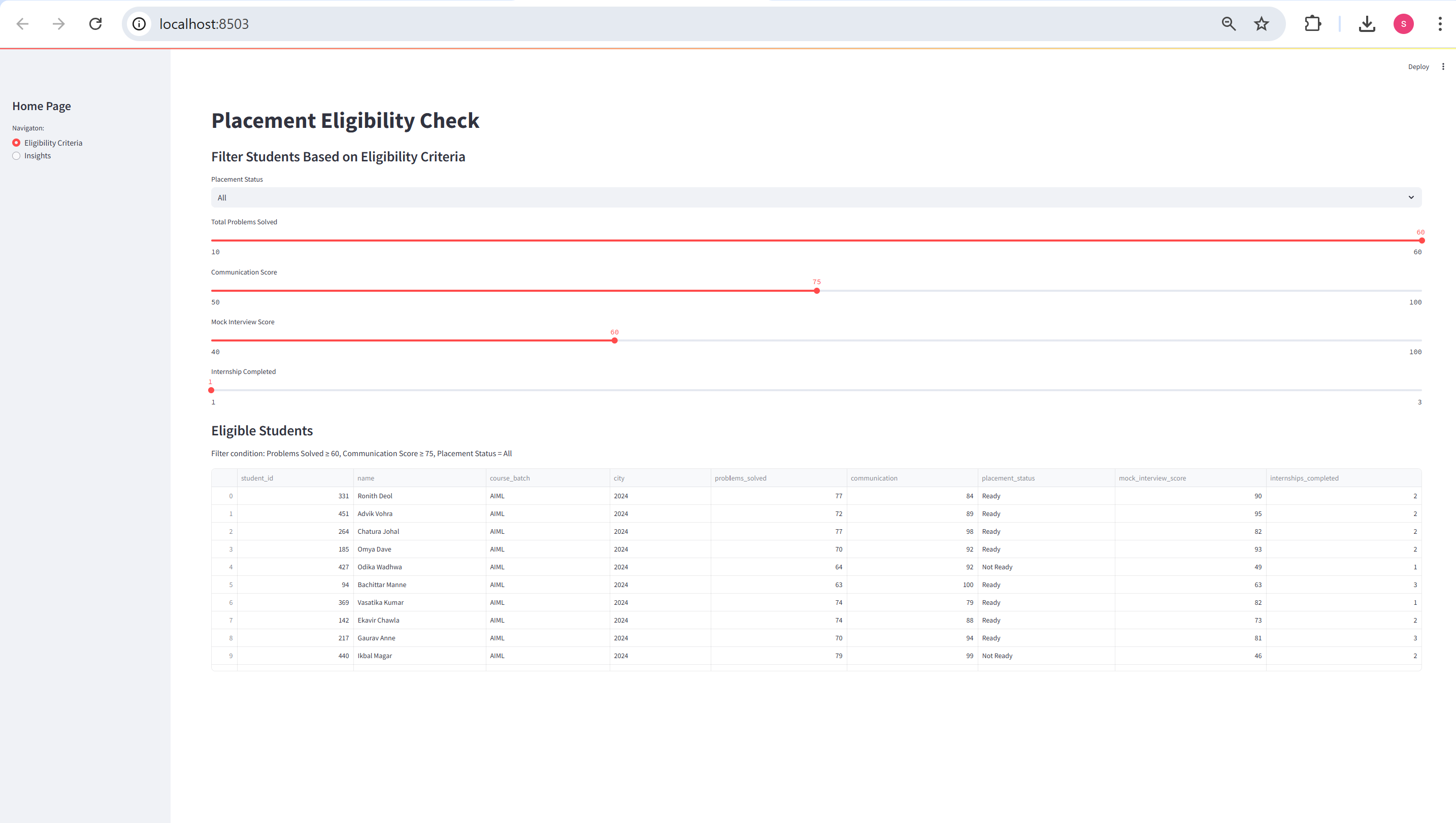
1.Data is generated using Faker and inserted into SQLite.  
2. Stream-lit UI accepts input filters for eligibility.  
3. SQL queries fetch and filter eligible candidates.  
4. Dynamic Results are displayed in a Stream-lit Application.

## Application Features:

1. Input-based student filtering (e.g., problem count > 50, communication > 75).  
2. Real-time display of eligible candidates.  
3. Stream-lit Application with key insights from SQL analytics.

## Application in Action:

1. Simple sidebar name as **Home Page** input for filter criteria.



1. Box showing all 10 SQL queries and Insights.

