Research Report

Population Growth Analysis – Pakistan.

Group Members:

- Abdul Lahad (007)
- Muhammad Razi (057)
- Uzair Hussain (085)
- Mirza Muhammad Abbas (101).

Course: Web Engineering

Table of Contents

1. Introduction

1.1 Project Overview

The *Pakistan City Population Data Visualization System* is a comprehensive full-stack web application designed to analyze and compare population data from Pakistan's 1998 and 2017 censuses. The application offers interactive charts, dynamic filtering options, and report generation tools to support researchers, policymakers, and students in understanding city-wise population growth and demographic trends.

1.2 Purpose

The core objectives of this system include:

- Providing an intuitive platform to explore and analyze city population trends across Pakistan.
- Enabling comparative analysis of city-wise growth rates within and across provinces.
- Offering downloadable, structured reports (PDF/CSV) for academic, administrative, or research-based decision-making.

1.3 Technologies Used

- **Frontend:** HTML5, CSS3, JavaScript
    - **Libraries:** Chart.js (for charting), PDF-Lib (for PDF generation), FileSaver.js (for CSV export)
- **Backend:** Node.js, Express.js, MongoDB, Mongoose
- **Deployment:** Localhost (scalable to cloud platforms such as Heroku, Render, or Vercel)

2. System Architecture

2.1 Frontend Structure

- **Dashboard:** Main interface displaying interactive data visualizations and a data table.
- **Filters:** Dropdown and range inputs to filter data based on province, population range, and growth rate.
- **Visualizations:**
    - **Bar Charts:** Population distribution by province.
    - **Line Graphs:** Growth trends over time.
- **Reporting Tools:** Export data and charts as downloadable PDF and CSV files.

2.2 Backend Structure

- **API Endpoint:** http://localhost:8000/api/data/getAll
- **Database:** MongoDB (major cities collection) stores structured city population data.
- **Routing:** Express.js routes handle frontend requests, retrieve data from the database, and send JSON responses.

2.3 Data Flow

1. The front end triggers a request to the backend API.
2. Backend processes the request and fetches data from MongoDB.
3. Retrieved data is formatted and sent back to the frontend for rendering and interaction.

3. Key Features

3.1 Interactive Data Visualization

- **Bar Charts:** Display of province-wise population data.
- **Growth Analysis:** Highlights top 10 fastest and slowest-growing cities.

- **Comparison Graphs:** Show 1998 vs. 2017 census data for each city.

3.2 Dynamic Filtering

- **Filter options include:**
    - Province (Punjab, Sindh, KPK, etc.)
    - Population Range (0 to 20 million)
    - Growth Rate (-100% to 500%)

3.3 Report Generation

- **PDF Reports: Automatically generated for:**
    - Top/Bottom 10 cities by population
    - Highest/Lowest growth rates
    - Province-wise summaries
- **CSV Export:** Allows users to download raw data for external analysis.

3.4 Responsive UI

- Fully functional on both desktop and mobile devices.
- Clean, minimalist design for ease of use.

4. Backend Implementation

4.1 API Routes

- **GET /api/data/getAll**: Retrieves all city records from the database.

4.2 Database Schema

{

Rank: Number,

City: String,

Province: String,

Population_2017_Census: Number,

Population_1998_Census: Number,

Change: Number

}

4.3 Error Handling

- **404**: Returned when no data matches the query.
- **500**: Triggered on internal server/database errors.

5. Frontend Implementation

5.1 Charts & Graphs

- Implemented using **Chart.js**
- Real-time updates based on filter changes
- Visual clarity maintained across different screen sizes

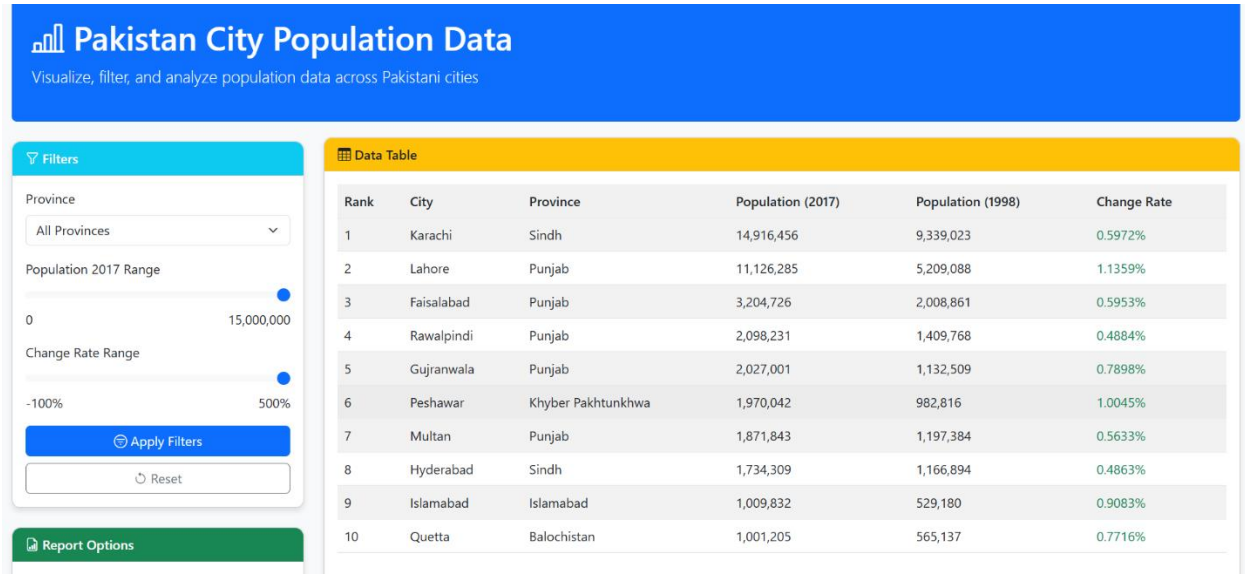
5.2 Data Table

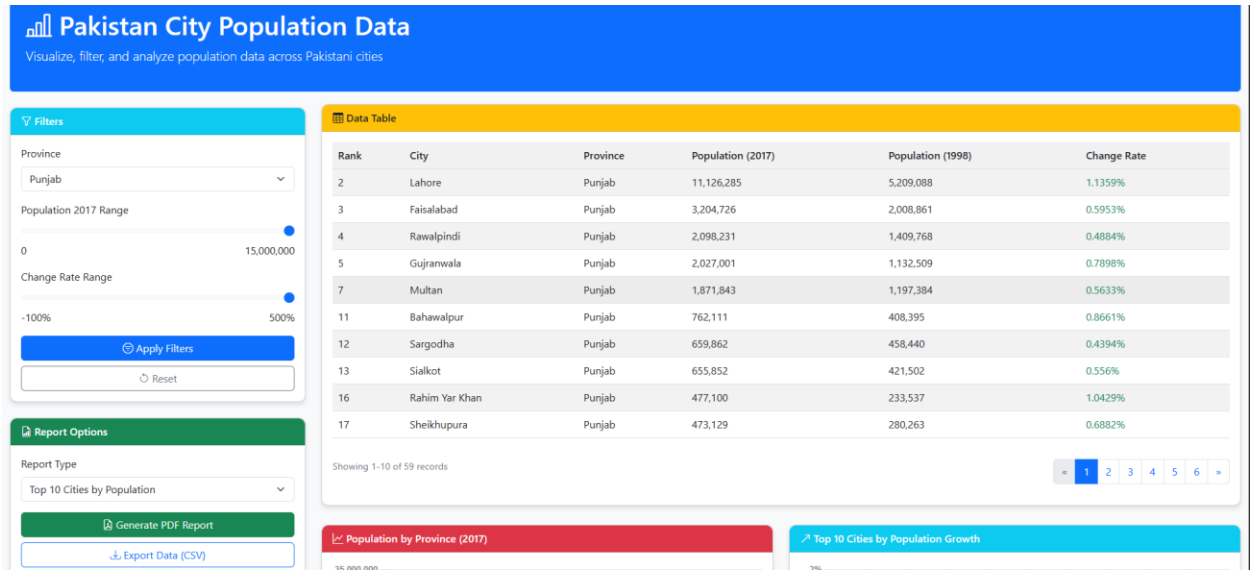
- Displays 10 records per page
- Sorting enabled on Rank, Population (1998/2017), and Growth Rate

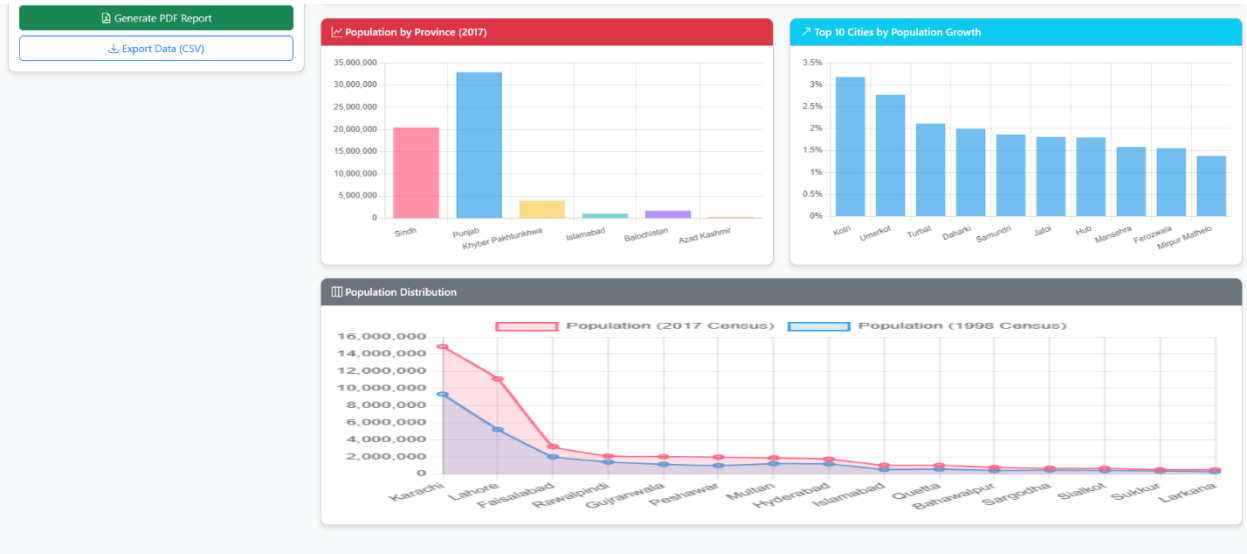
5.3 Export Functionalities

- **PDF-Lib**: Generates custom reports based on user-selected criteria.
- **FileSaver.js**: Enables downloading of filtered data in CSV format.

6. Demo & Visuals







- **Dashboard View**: Visualizes population and growth data with clear charts and filters.
- **Filtering Data**: Users can fine-tune results based on their region of interest.
- **Generated PDF Reports**: Structured, styled documents exportable for offline use.

7. Challenges & Solutions

Challenge	Solution
Large dataset rendering	Implemented client-side pagination in the data table.
Real-time chart updates	Integrated live data binding with Chart.js.
Structured PDF generation	Utilized PDF-Lib for layout customization.

8. Future Enhancements

- **Geographical Maps**: Visual plotting of cities on an interactive Pakistan map.
- **User Accounts**: Enable users to save filters and download history.
- **Historical Data Support**: Include older census records for extended analysis.
- **Automated Updates**: Implement scheduled scripts or APIs for updating census datasets.

9. Conclusion

9.1 Impact

This system bridges the gap between raw census data and user-friendly visualization. Its impact includes:

- Aiding **policymakers** in making data-informed decisions on urban development.
- Assisting **researchers and students** in performing comparative studies.
- Providing a foundation for **expanding similar tools** for other national datasets.

9.2 Final Thoughts

The Pakistan City Population Data Visualization System is not only a technical accomplishment in full-stack development but also a step toward using data meaningfully in real-world scenarios. With potential for scale and real-time integration, this system could become a key tool in national demographic analysis.