

Population Report System

Checklist 2



|  |  |
| --- | --- |
| Name | ID |
| Syed Abdul Samad Zaidi | 40692317 |
| Muhammad Ali Khalid | 40692222 |
| Muhammad Taha Khalid | 40692271 |
| Hamza Shahid | 40692218 |

**https://github.com/Hamzashahid08/population\_report\_generator**

1. **GitHub issue**

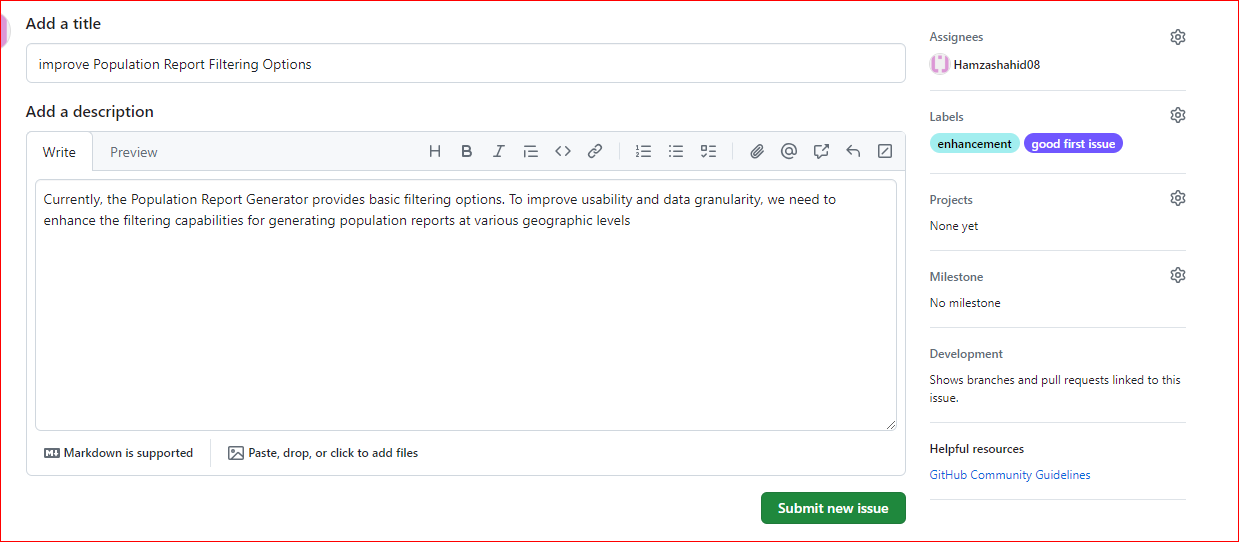
**Enhancing Population Report Filtering Options**

**What’s the Goal?** We want to improve how users filter population reports in our system. Right now, it’s basic, but we can make it better!

1. **User Interface (UI) Improvements:**

* Make the screen where users choose filters look nicer.
* Add options like continents, regions, or specific countries.
* Use dropdown menus or checkboxes so users can pick easily.

1. **Behind the Scenes:**
   * Change the way our program works:
     + When users pick filters, our program should adjust accordingly.
     + We’ll also update how we get data from the database based on the chosen filters.



1. **User Stories for Project Requirements**

* **View Population Trends**

As a user, I want to see how populations have changed over the past decade. This helps me understand demographic shifts over time.

* **Export Reports**

As a user, I’d like to export population reports to CSV format. This way, I can analyze the data in tools like Excel.

* **Filter Reports by Year**

As a user, I need to filter population reports by specific years. It makes comparing data across different time periods easier.

* **Top Populated Cities**

As a user, I want a list of the most populated cities. This helps me understand densely populated areas.

* **Visualize Data**

As a user, I’d like to see population data in charts. Visualizing trends makes it easier to understand demographic patterns.

* **Language Speakers Statistics**

As a user, I’m curious about global language diversity. Show me statistics on language speakers.

* **Compare Countries**

As a user, I want to compare population data between countries. It helps me see worldwide demographic differences.

* **Age Demographics**

As a user, I’m interested in age groups. Show me demographics related to different age ranges.

* **Integration with External Data**

As a user, I’d like to combine our data with external sources (like weather APIs). This enriches our demographic analysis.

* **Mobile-Friendly Interface**

As a user, I need a smartphone-friendly interface. I want to access population reports on the go.

1. **Zube.io**

By integrating our project with Zube.io, we’ve streamlined our project management. Our Kanban board helps us keep track of tasks, collaborate with team members, and smoothly move from backlog to completion. It’s a great way to ensure our development process stays on track

**Project Integration:**

* Integrated the GitHub repository ("population\_report\_generator") with Zube.io for seamless task tracking and management.

**Kanban Board Configuration:**

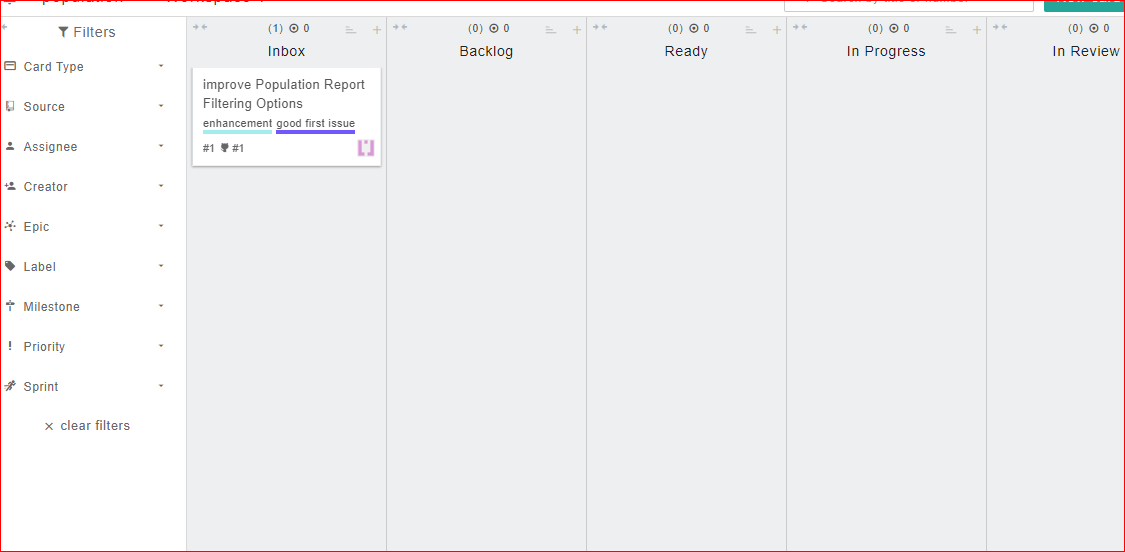
* Configured Kanban boards in Zube.io to reflect project tasks, bugs, and feature requests from GitHub issues.

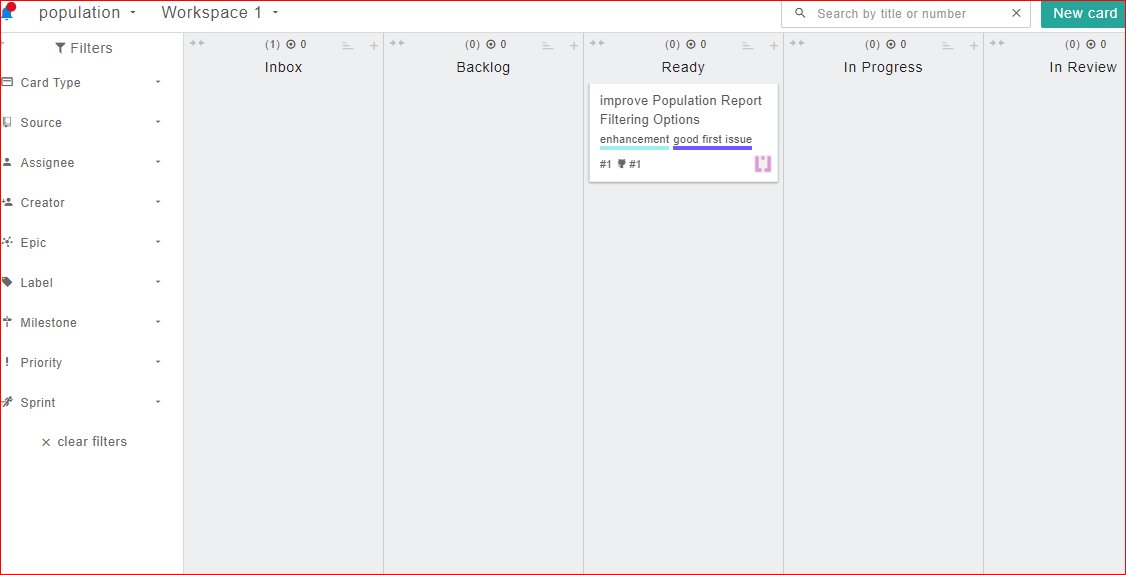
**Workflow Visualization:**

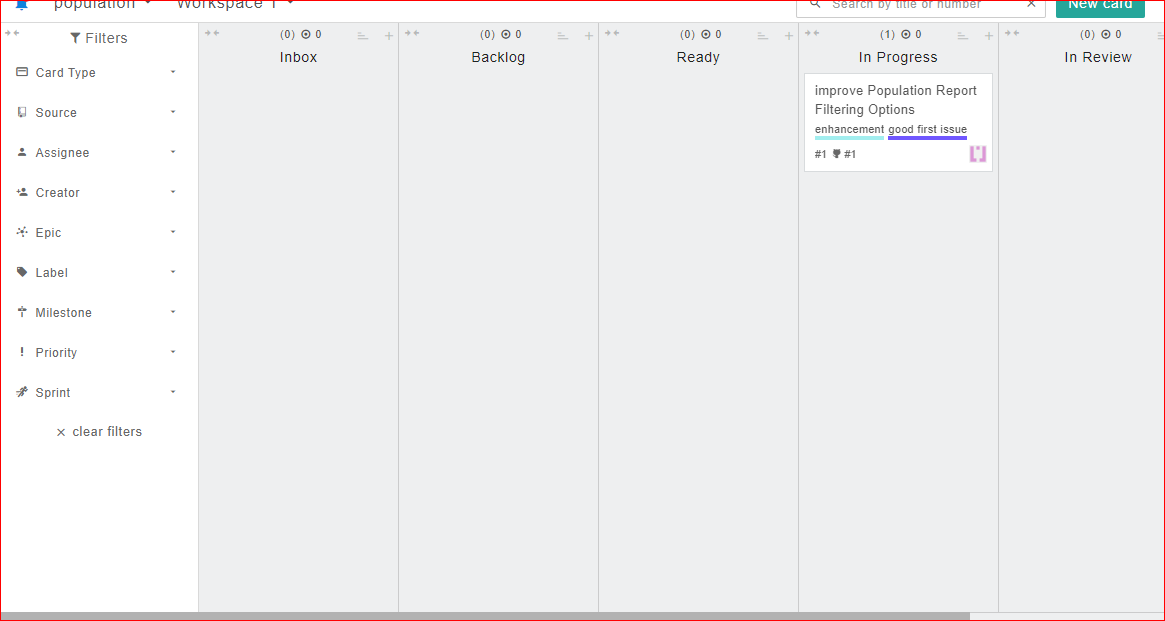
* Utilized Zube.io's Kanban boards to visualize the project workflow, from backlog to completion, ensuring transparency and accountability.

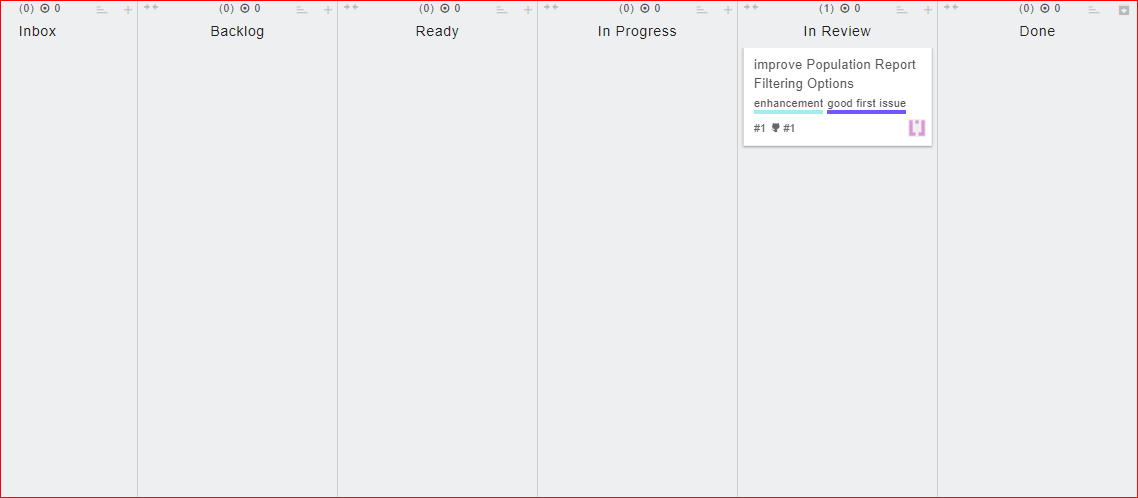
**Task Management Efficiency:**

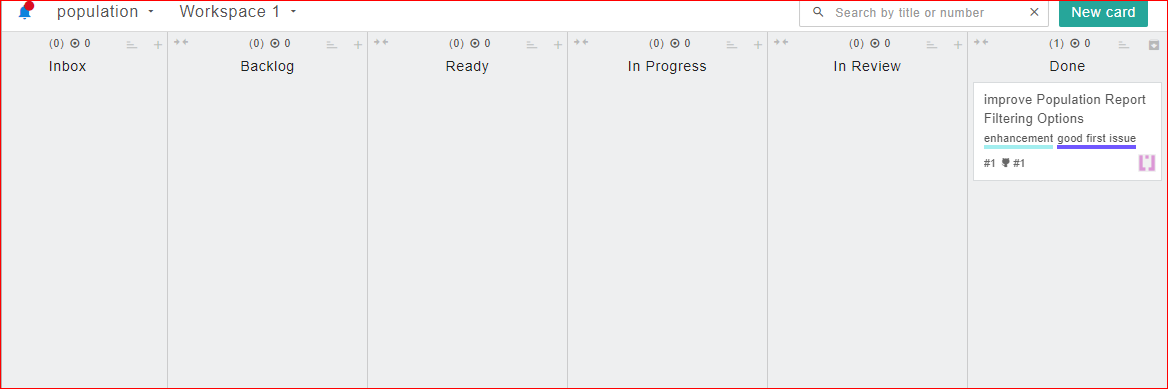
* Improved project efficiency by tracking tasks through Zube.io, enhancing collaboration and progress monitoring among team members.

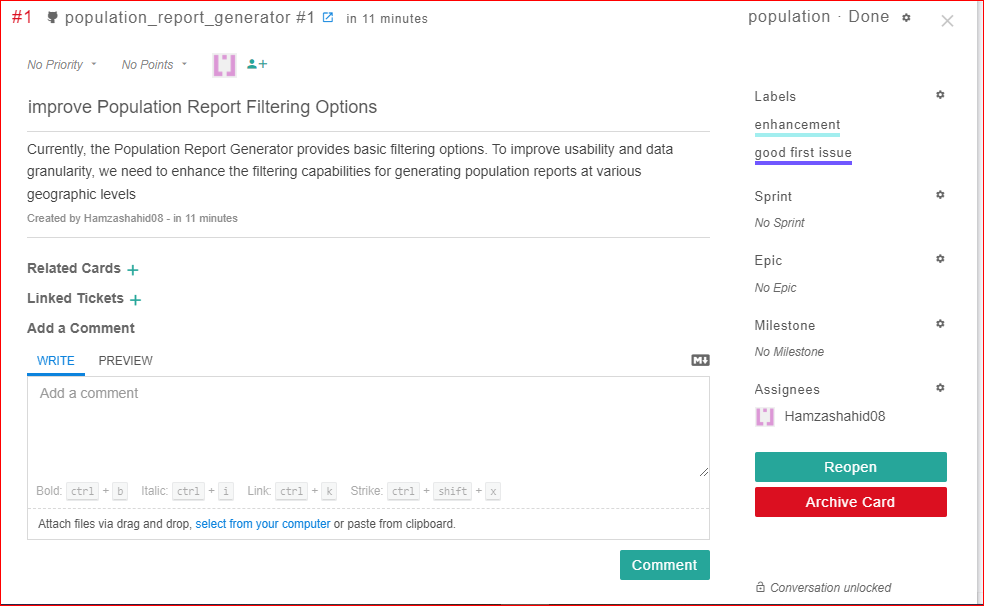












1. **Use Cases:**

**1. Use Case: View Countries by Population**

* **Actors:** User
* **Preconditions:** System is running and database connection is established.
* **Triggers:** User selects "View Countries by Population" option from the main menu.
* **Flow:**
  1. User selects "View Countries by Population".
  2. System retrieves a list of countries ordered by population.
  3. System displays the list of countries with their respective populations.
* **Postconditions:** User views countries sorted by population.

**2. Use Case: View Cities by Population**

* **Actors:** User
* **Preconditions:** System is running and database connection is established.
* **Triggers:** User selects "View Cities by Population" option from the main menu.
* **Flow:**
  1. User selects "View Cities by Population".
  2. System retrieves a list of cities ordered by population.
  3. System displays the list of cities with their respective populations.
* **Postconditions:** User views cities sorted by population.

**3. Use Case: View Top N Populated Cities**

* **Actors:** User
* **Preconditions:** System is running and database connection is established.
* **Triggers:** User selects "View Top N Populated Cities" option from the main menu.
* **Flow:**
  1. User selects "View Top N Populated Cities".
  2. User inputs the number of cities (N).
  3. System retrieves and displays the top N cities ordered by population.
* **Postconditions:** User views top N cities sorted by population.

**4. Use Case: Generate Language Speakers Statistics**

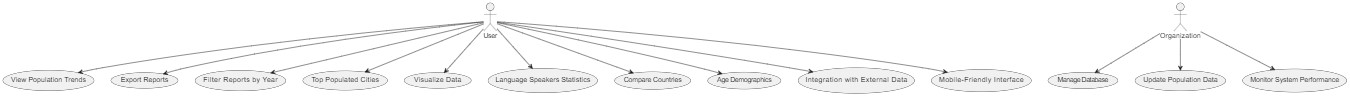
* **Actors:** User
* **Preconditions:** System is running and database connection is established.
* **Triggers:** User selects "Generate Language Speakers Statistics" option from the main menu.
* **Flow:**
  1. User selects "Generate Language Speakers Statistics".
  2. System retrieves language populations from the database.
  3. System calculates and displays statistics on language speakers.
* **Postconditions:** User views language speakers statistics.

**5. Use Case: Exit System**

* **Actors:** User
* **Preconditions:** User is interacting with the system.
* **Triggers:** User selects "Exit" option from the main menu.
* **Basic Flow:**
  1. User selects "Exit".
  2. System terminates the program and closes database connection.
* **Postconditions:** System exits gracefully.

1. **Diagrams**

**Use case Diagram:**



**Activity Diagram**

