# **Population Stats Report - Project Documentation**

## **1. Project Overview**

The **Population Stats Report** is a C# console-based application that connects to a MySQL database and provides statistical reports on:

* Countries sorted by population
* Top cities in a continent
* Language speaker statistics

This project follows Agile methodology using **Scrum**, with task management done via **Zube.io** and version control using **GitHub**.

## **2. Technology Stack**

* **Programming Language:** C#
* **Database:** MySQL (Wamp)
* **IDE:** Visual Studio
* **Version Control:** GitHub
* **CI/CD:** GitHub Actions
* **Project Management:** Zube.io

## **3. Team Roles & Responsibilities**

### **Product Owner: Aoun Bilal**

* Managed project scope and requirements
* Ensured timely delivery and quality
* Communicated with stakeholders

### **Scrum Master: Tayyab Raza**

* Facilitated daily stand-ups and sprint meetings
* Ensured Agile practices were followed
* Removed blockers for the team

### **Developer 1: Adil Ghuman**

* Implemented **MySQL queries** and database integration
* Developed the DatabaseHelper.cs class

### **Developer 2: Salman Riaz**

* Implemented **C# application logic**
* Developed Program.cs for handling user input and output

## **4. Project Workflow**

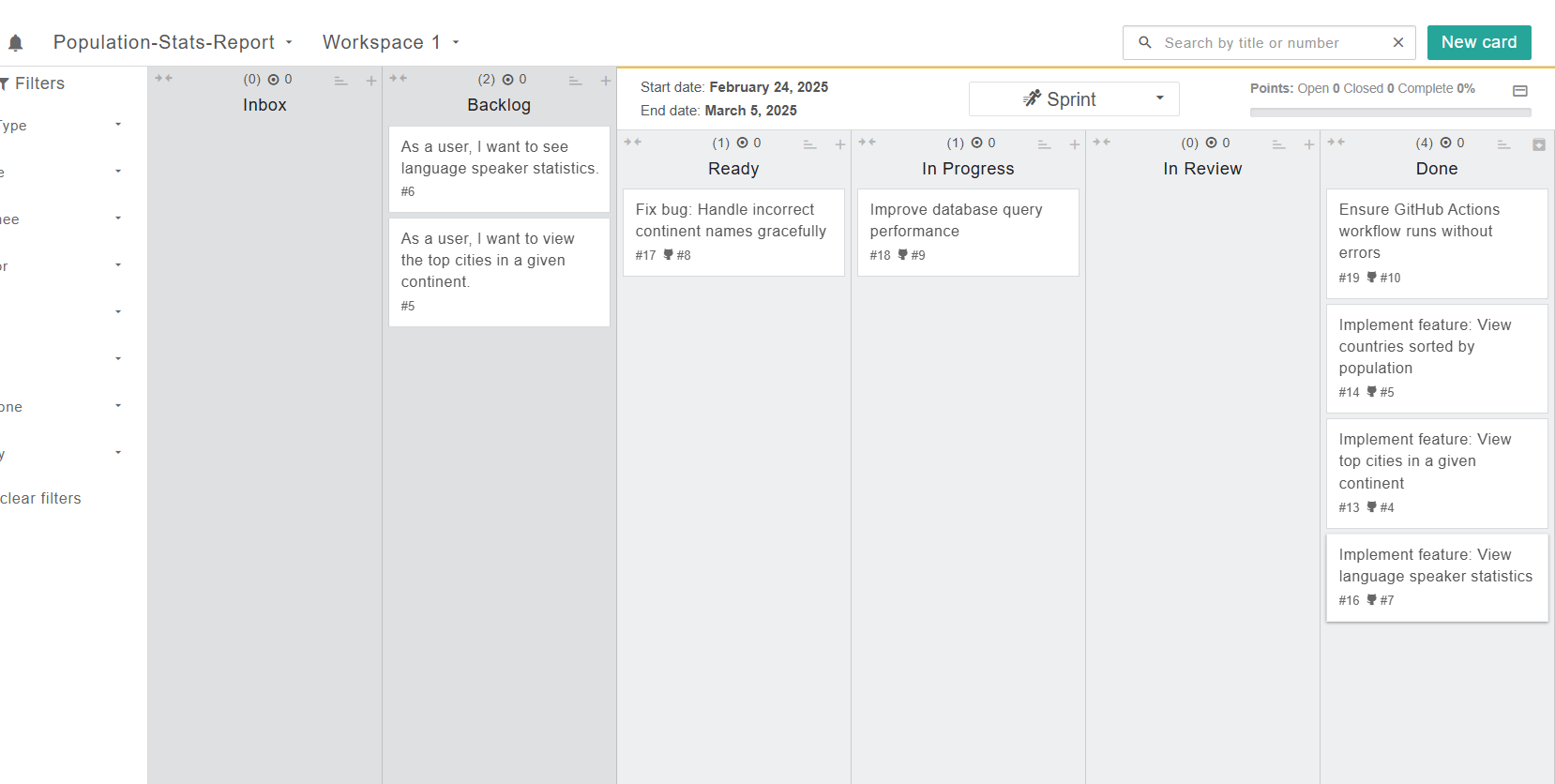
The project followed an **Agile Scrum approach**, with task tracking done in **Zube.io**.

### **4.1 Sprint Planning & Task Assignments**

1. Defined user stories and created issues in GitHub/Zube.io

2. Assigned tasks to developers based on roles

3. Tracked progress using the Kanban board

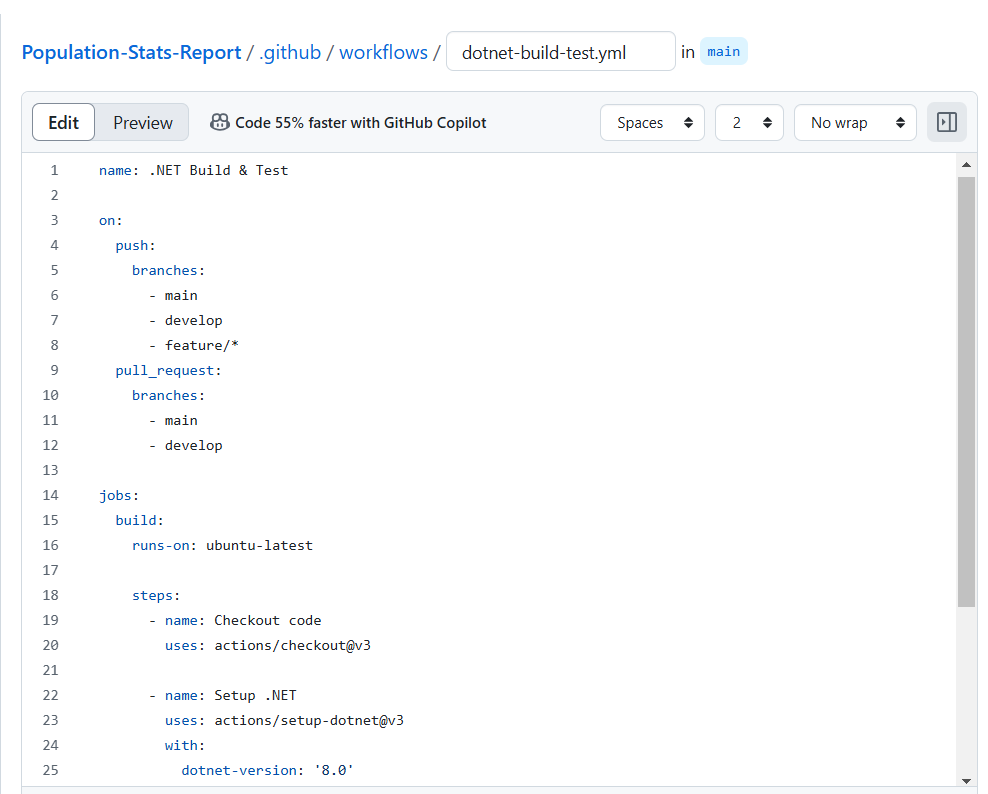


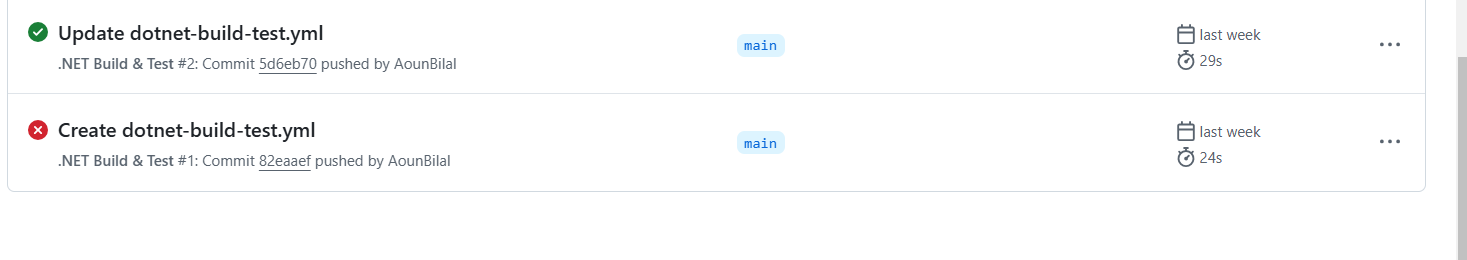
### **4.2 Daily Stand-ups & Development Process**

1. Conducted stand-ups to discuss progress and blockers

2. Developers worked on assigned features and pushed code to GitHub

3. CI/CD pipeline (GitHub Actions) verified code integrity



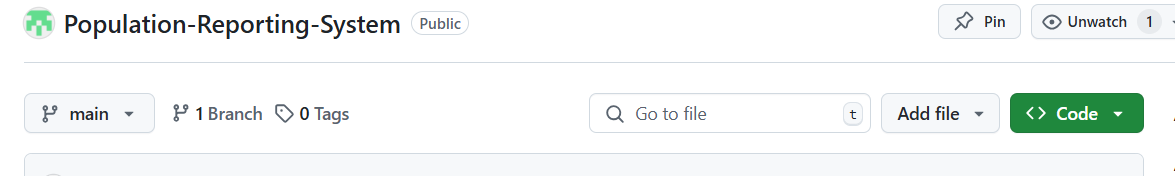
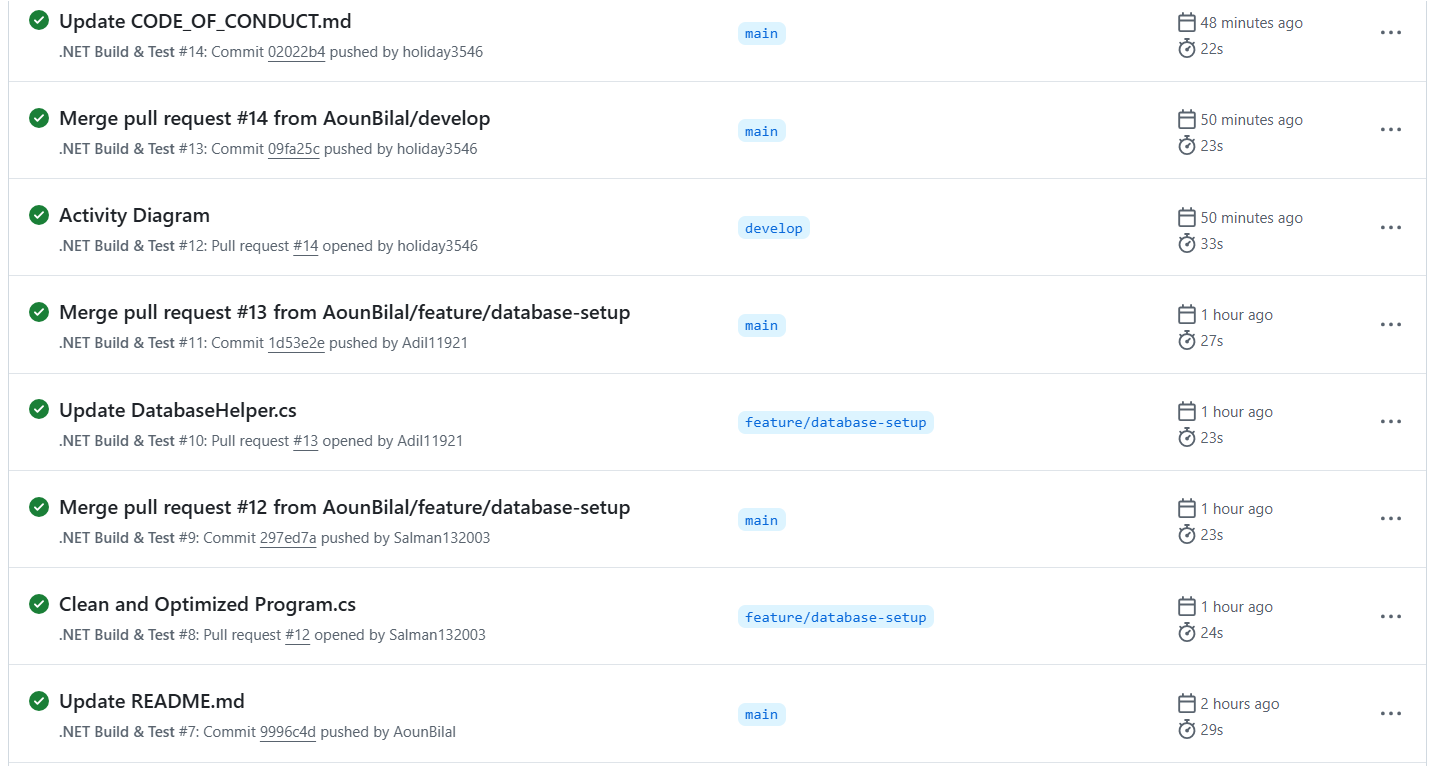


### **4.3 Sprint Review & Finalization**

1. Verified all reports were working correctly

2. Ensured GitHub Actions tests passed successfully

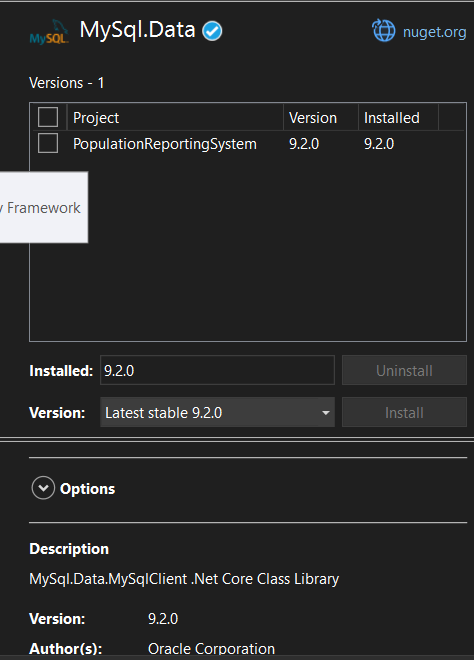
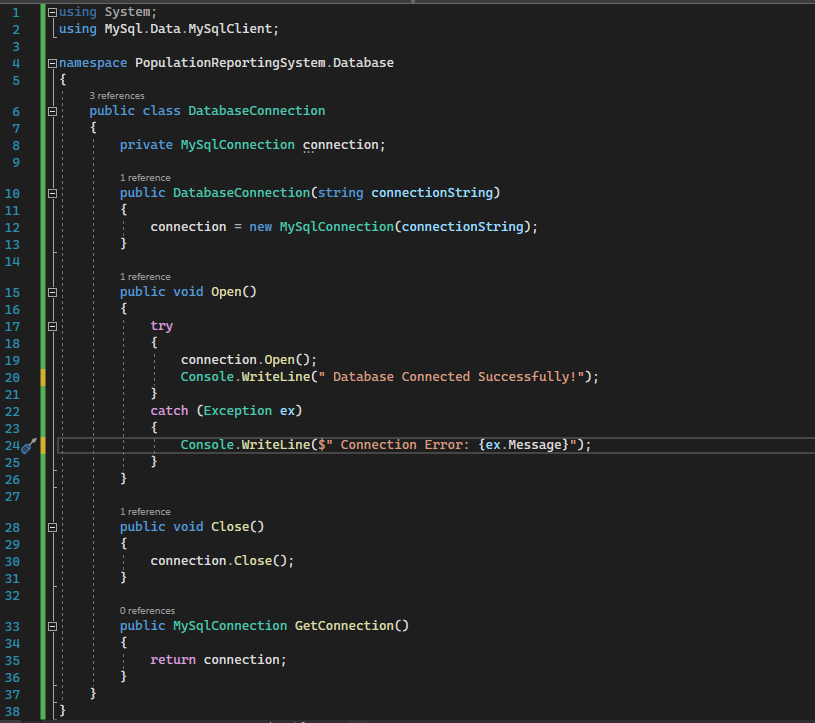
3. Prepared final submission files

## **5. Implementation Details**

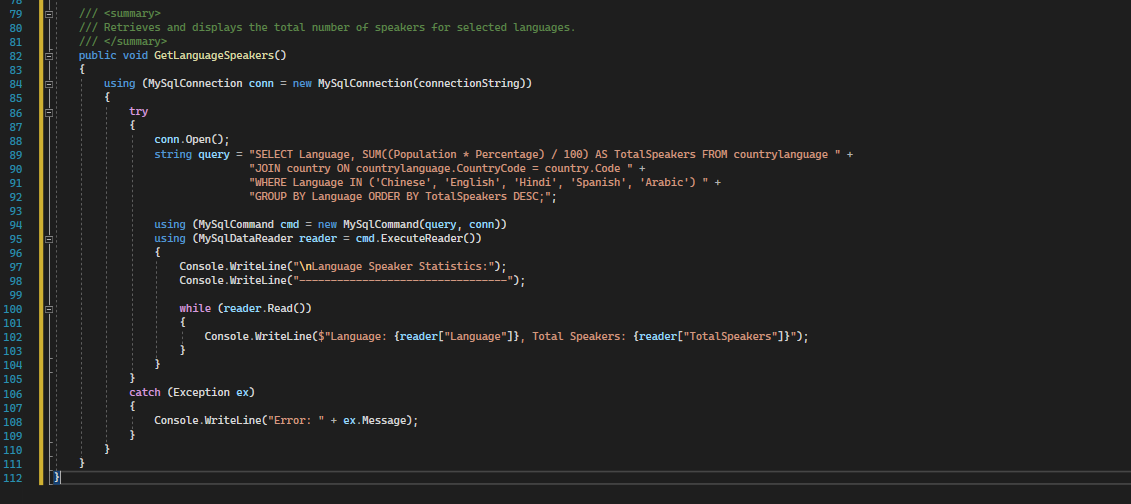
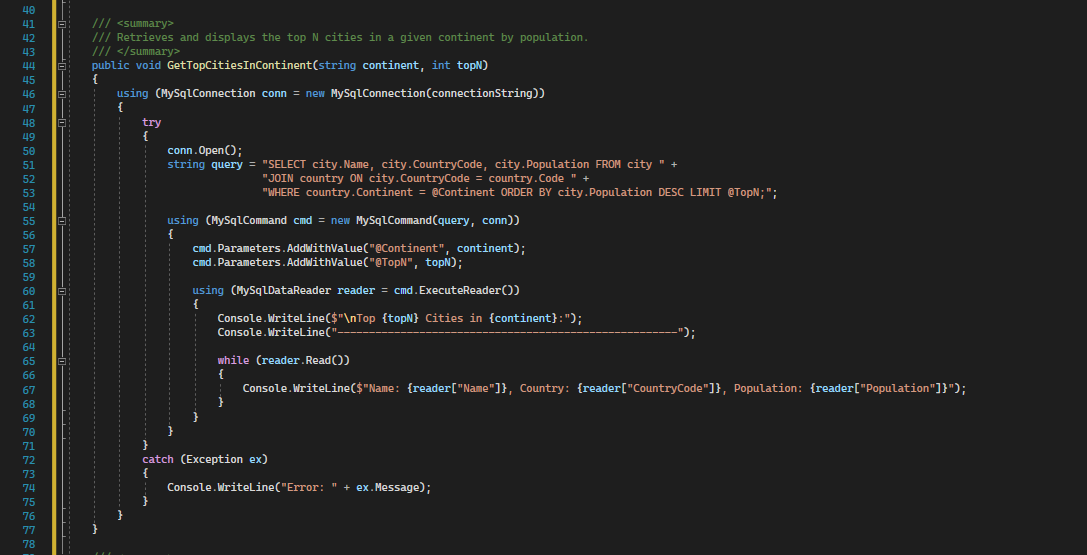
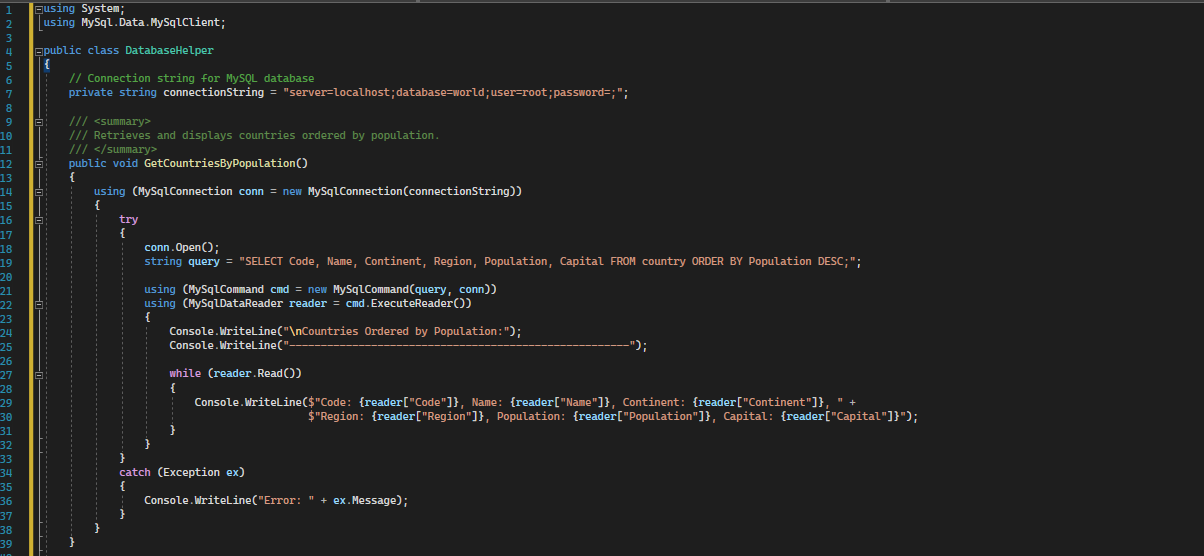
### **5.1 MySQL Database Integration**

* Connected to MySQL using **MySql.Data.MySqlClient**
* Implemented SQL queries to fetch population data

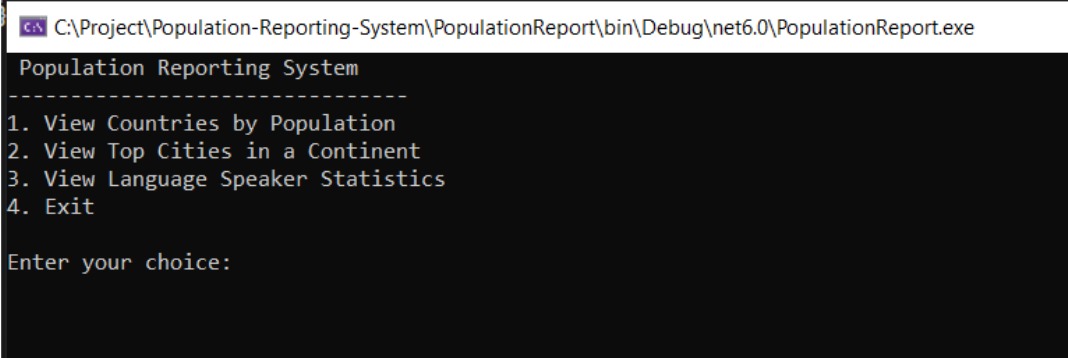
  
  

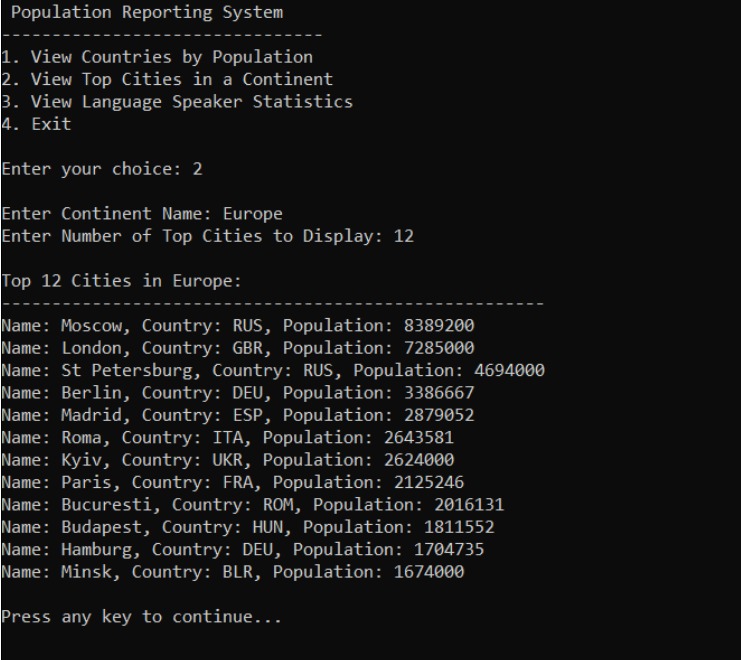
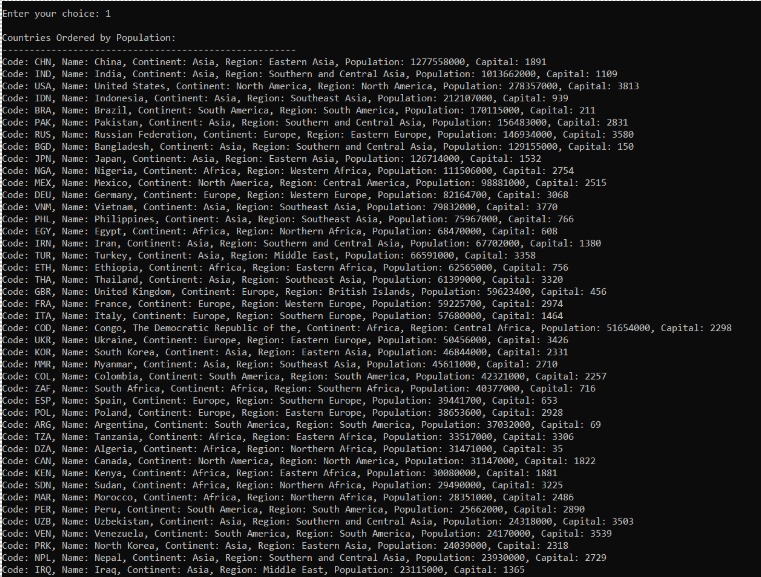

### **5.2 C# Code Implementation**

* DatabaseHelper.cs: Handles database queries



**Outputs**



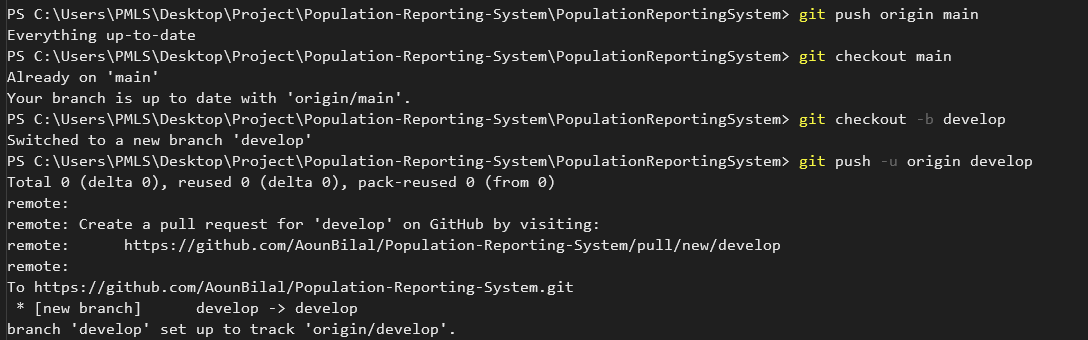
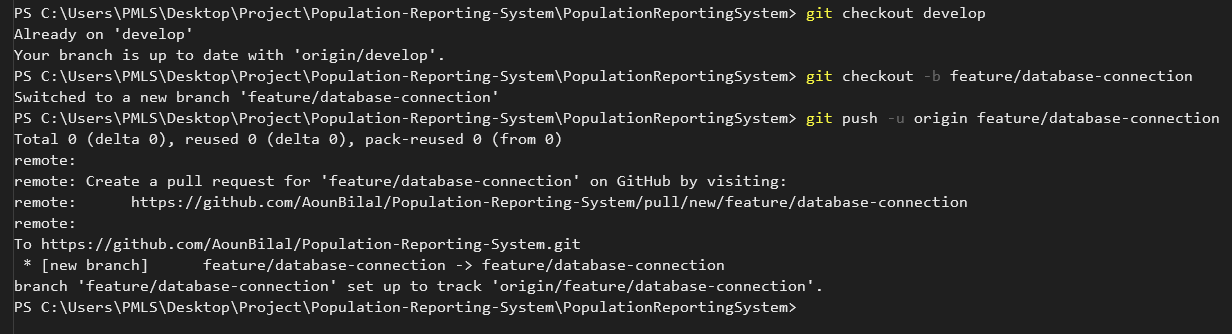
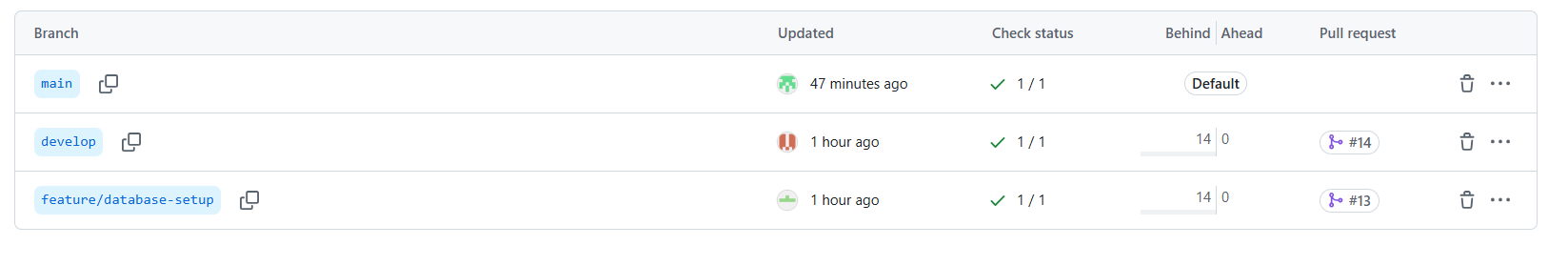
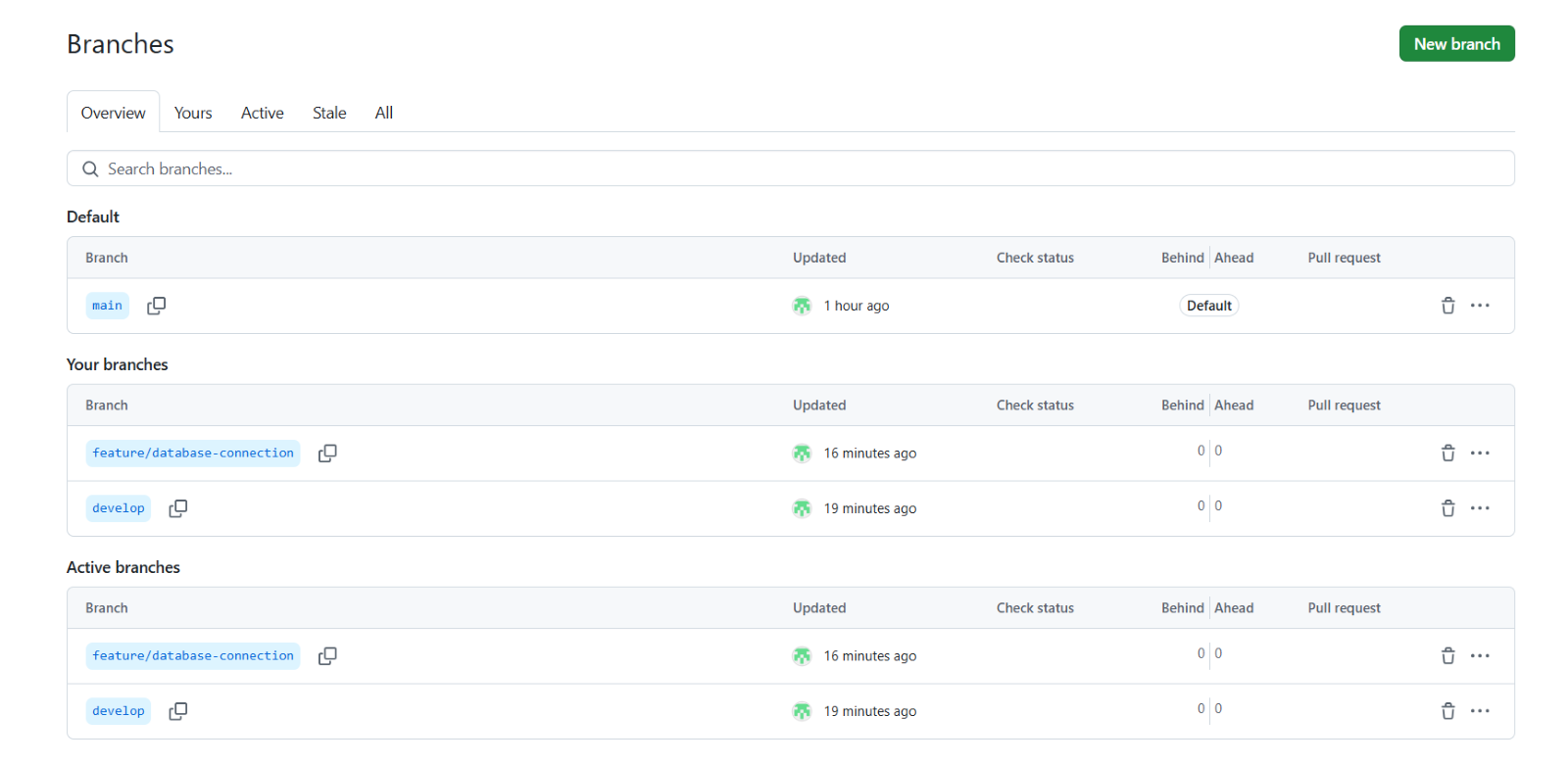


**5.3 GitHub Repository & Branching Strategy**

1. Followed GitHub Flow (main, develop, feature/\* branches)

2. Used Pull Requests for merging features

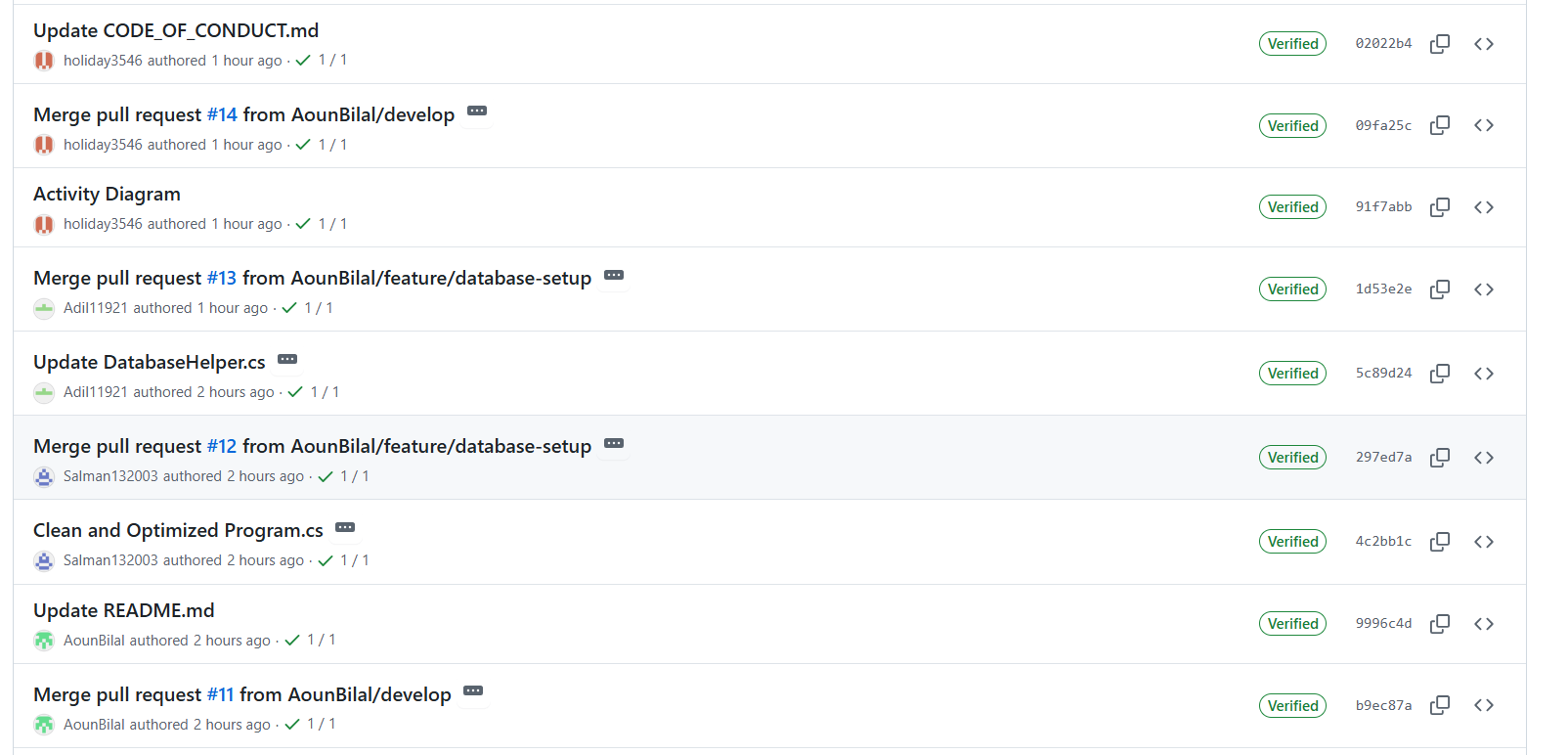
3. Tracked commits and contributions

### **5.4 GitHub Actions (CI/CD Pipeline)**

1. Automated build and test process

2. Ensured every commit was verified

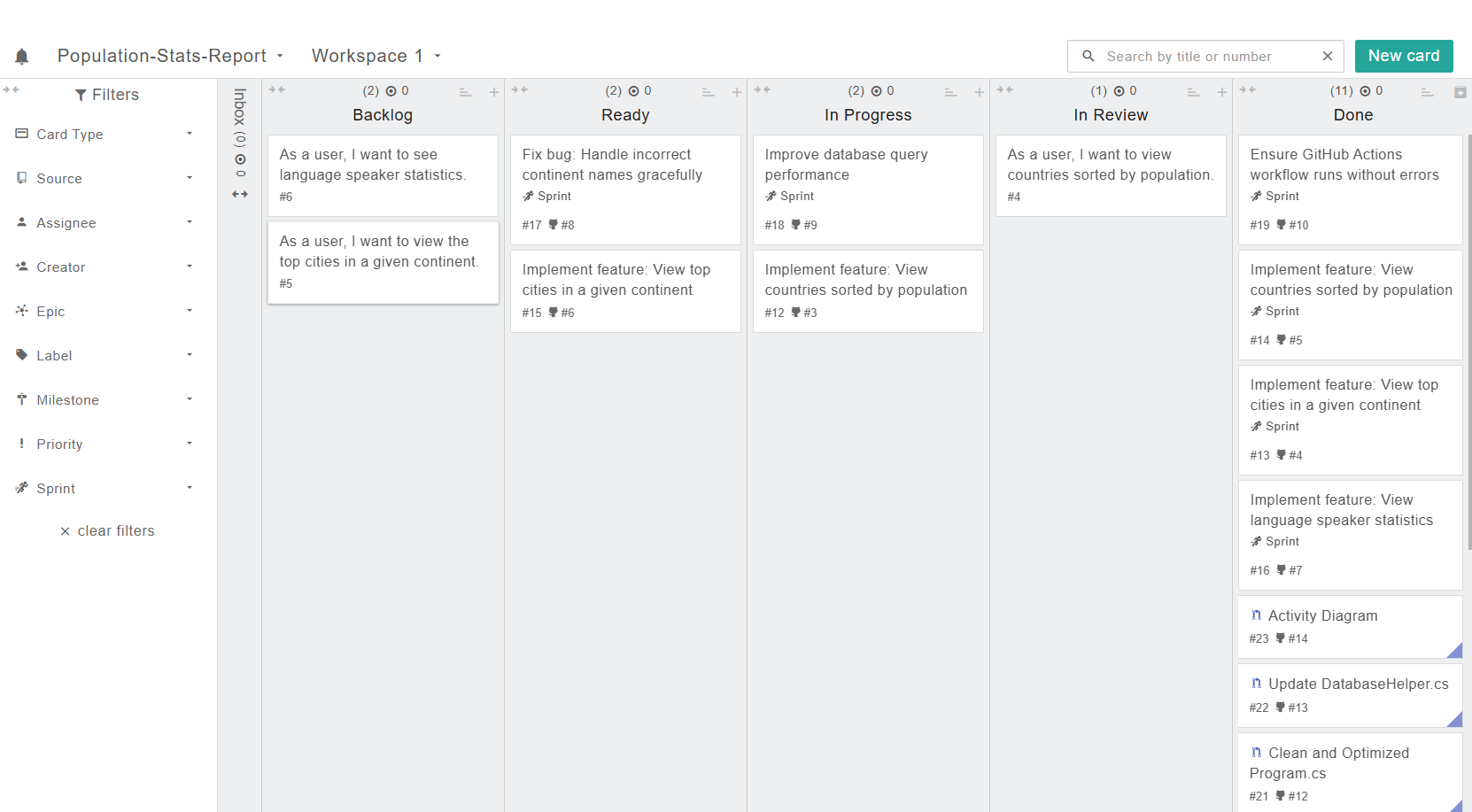
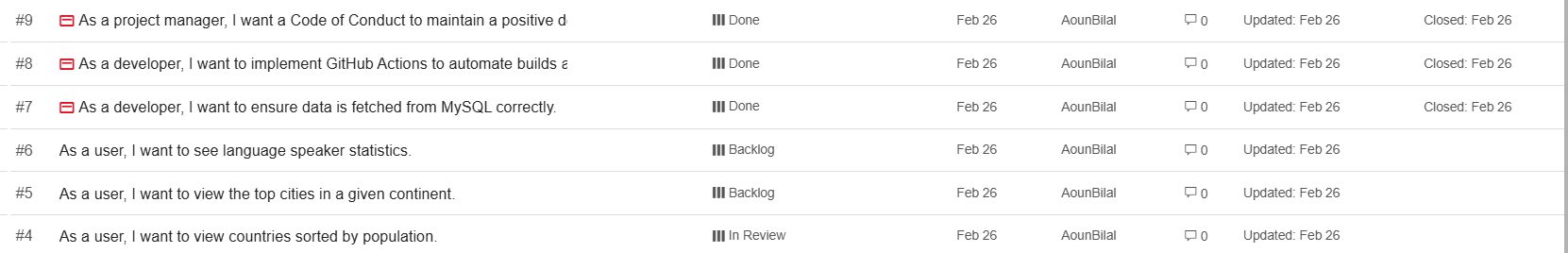


### **5.5 Task Management in Zube.io**

1. Created user stories and tasks

2. Moved completed tasks to "Done"

3. Ensured all requirements were tracked



## **6. Final Submission Details**

### **6.1 GitHub Repository Link**

**GitHub Repo:** <https://github.com/AounBilal/Population-Stats-Report.git>

### **6.2 Zube.io Kanban Board Link**

**Zube.io Board:** <https://zube.io/group/population-stats-report/w/workspace-1/kanban>

## **7. Conclusion**

The **Population Stats Report** project successfully implemented a C# console application connected to a MySQL database, following Agile Scrum practices. The project was well-organized with proper GitHub version control, CI/CD automation, and Zube.io task management.