Project Name: Process Gantt Chart Visualizer

Contributor: Rekula Dakshi Venkata Chaudhary

Roll No.: 21EC10056

Code Github link:

Project Overview:

This project focuses on creating a comprehensive and user-friendly system for simulating, analyzing, and visualizing process scheduling algorithms, commonly used in operating systems. The aim is to provide an educational and practical tool that helps users understand how different scheduling algorithms work, their performance, and their impact on system efficiency.

Key Features:

1. Implementation of Process Scheduling Algorithm:

- The system supports a variety of common process scheduling algorithms such as First-Come-First-Serve (FCFS), Shortest Job First (SJF), Round Robin (RR), and Priority Scheduling.

- Each algorithm has been implemented to demonstrate its unique method of managing process execution order, considering factors like process arrival time, burst time, and priority.

2. Interactive Visualization:

- A Gantt chart visualization is employed to display the sequence and timing of processes being executed by the scheduler.

- This graphical representation provides a clear view of the process execution timeline, making it easier for users to understand the scheduling process and the distribution of CPU time among various processes.

3. Statistical Analysis:

- The tool includes a feature for statistical analysis of the scheduling results. Metrics such as average waiting time, turnaround time, and CPU utilization are calculated and displayed.

- This analysis helps users compare the efficiency of different scheduling algorithms under various scenarios, providing insights into which algorithms perform best under specific conditions.

4. User-Friendly Interface:

- The system is designed with a user-friendly interface, allowing users to easily input process details and select the scheduling algorithm they wish to simulate.

- The visual and statistical outputs are presented in a clear and intuitive manner, making the tool accessible even to those with limited background knowledge in operating systems.

5. Real-Time Simulation:

- The simulation feature allows users to observe the execution of processes in real-time. As the simulation runs, the Gantt chart and statistics are updated dynamically, providing a real-time view of how the scheduling algorithm is managing the processes.

Technical Implementation:

- Frontend: Developed using HTML, CSS, and JavaScript to create a responsive and interactive user interface.

- Backend Logic: Implemented in JavaScript, ensuring efficient simulation of scheduling algorithms and real-time updates of visualizations and statistics.

- Data Handling: JSON files are used to store and manage process data and scheduling results, facilitating easy manipulation and retrieval of data for simulation and analysis.

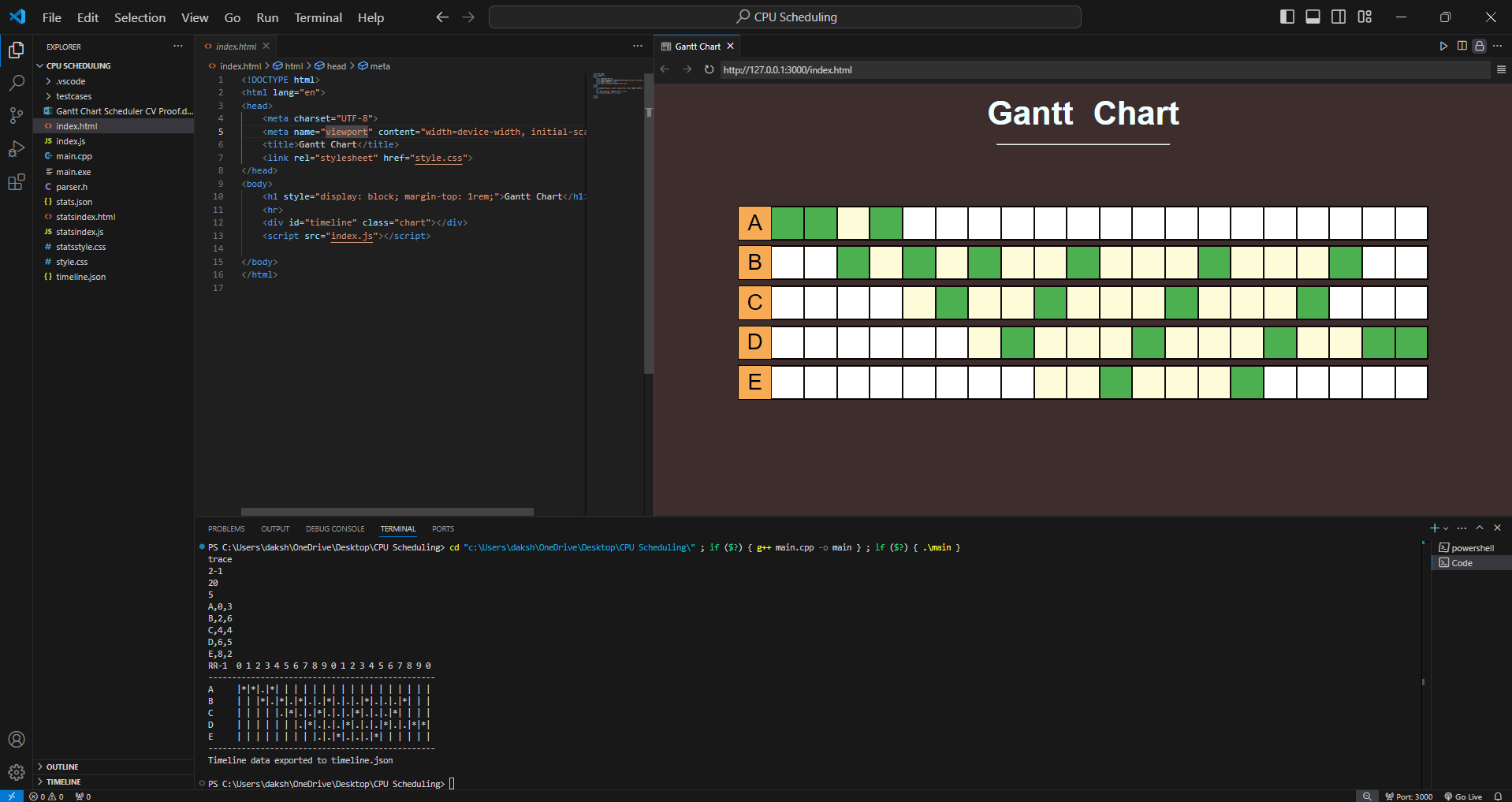
Project Outcomes:

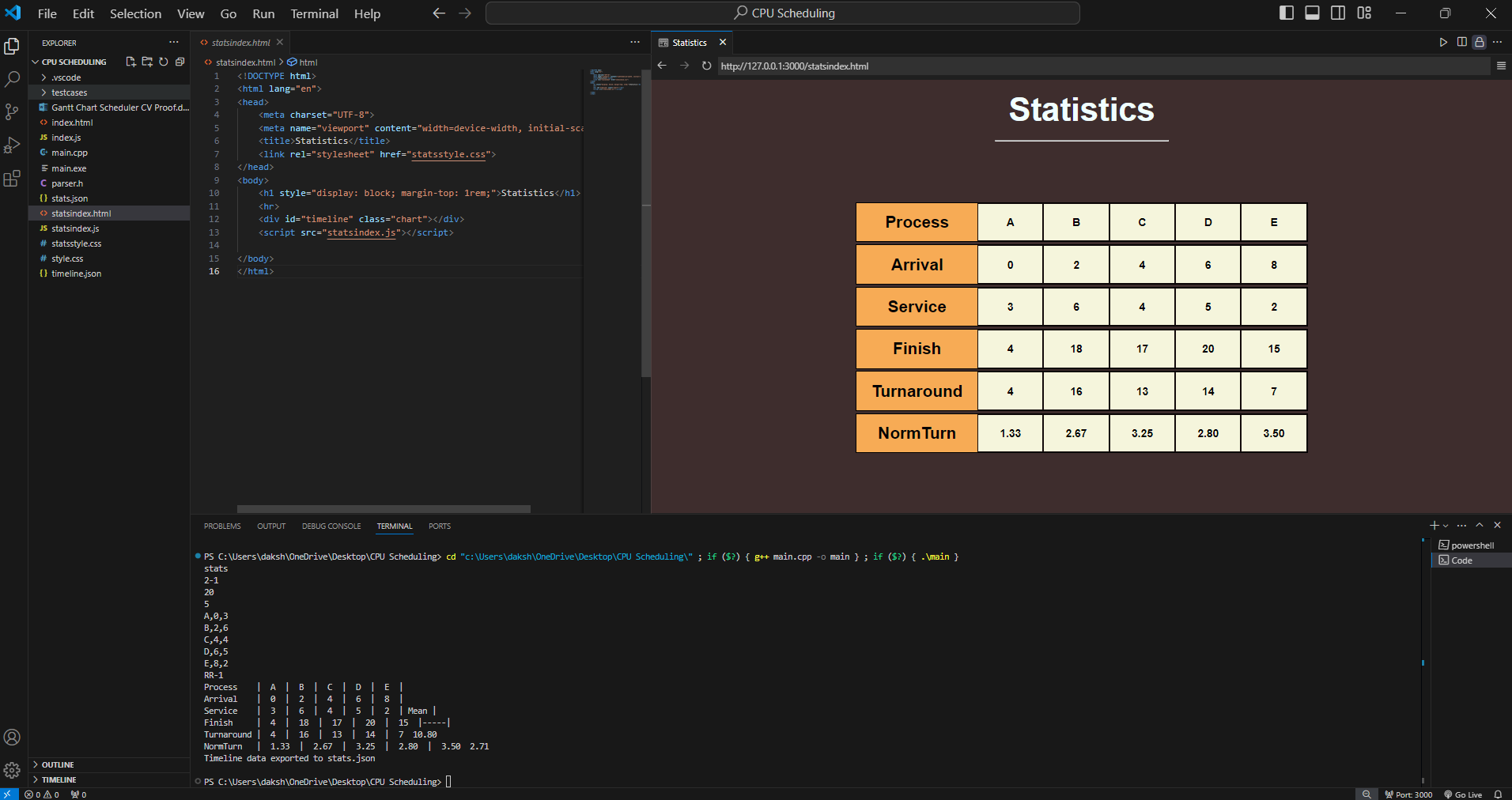
The Process Scheduler and Visualizer project successfully demonstrates how various scheduling algorithms function and their impact on system performance. It serves as an educational tool for students and professionals, providing hands-on experience with process scheduling concepts. By visualizing process execution and providing statistical insights, the tool enhances understanding and helps users make informed decisions about which scheduling strategies to employ in different scenarios.

Conclusion:

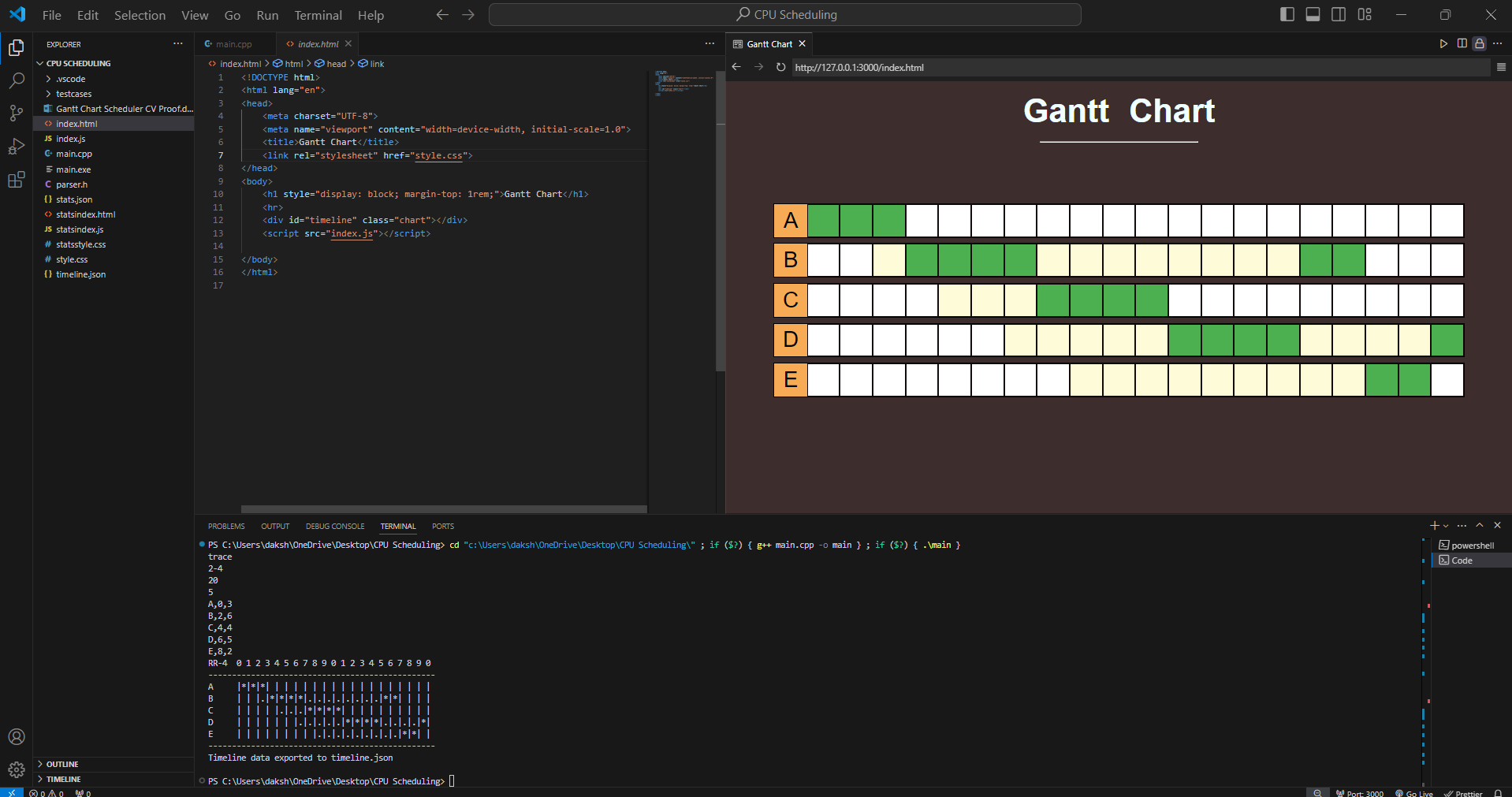
The Process Scheduler and Visualizer project effectively bridges the gap between theoretical understanding and practical application of process scheduling algorithms. Its interactive nature, combined with robust visualization and statistical analysis features, makes it a valuable tool for both learning and practical application in system optimization.

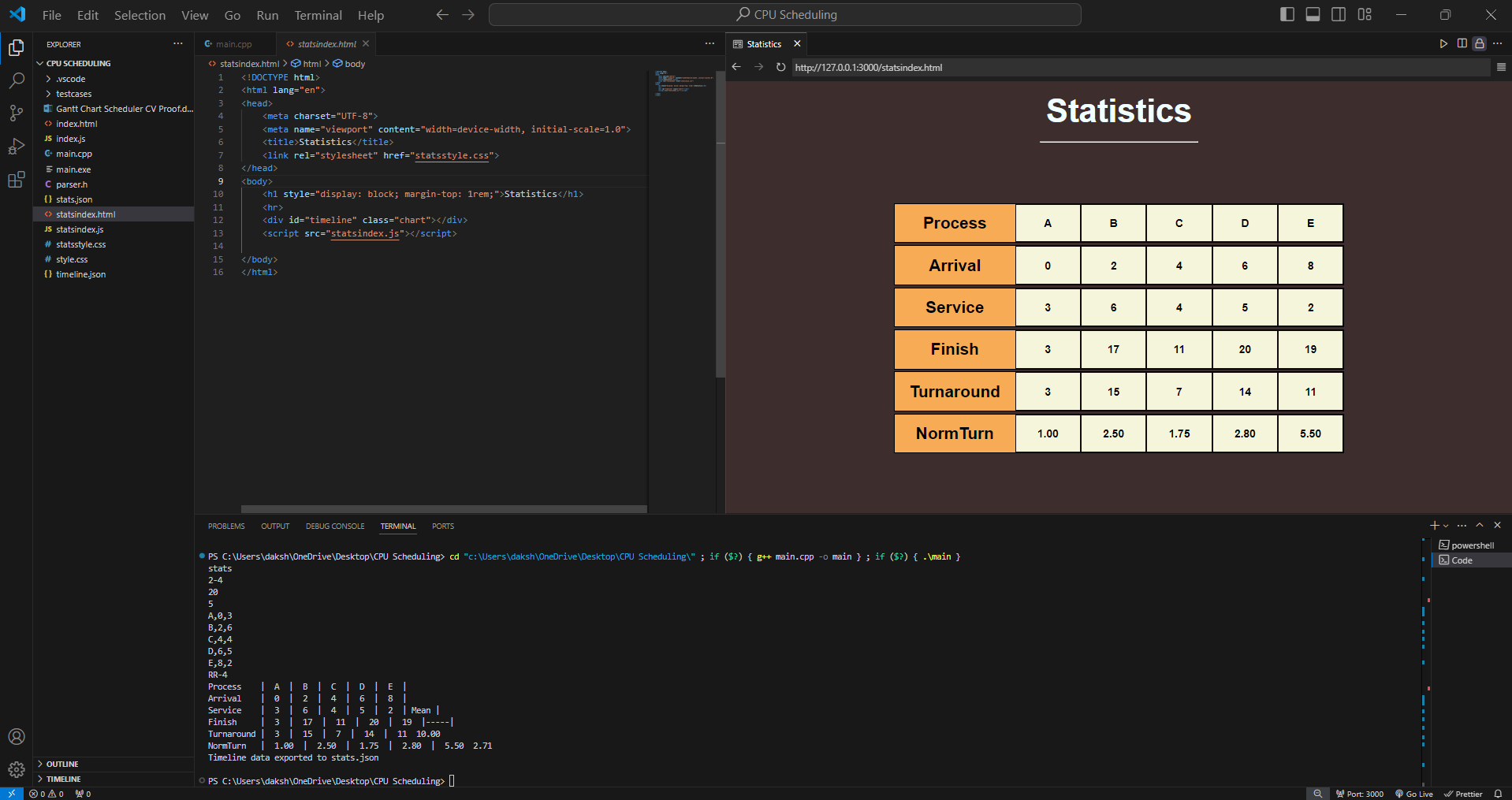
Trace and Statistics of an Example schedule 1:



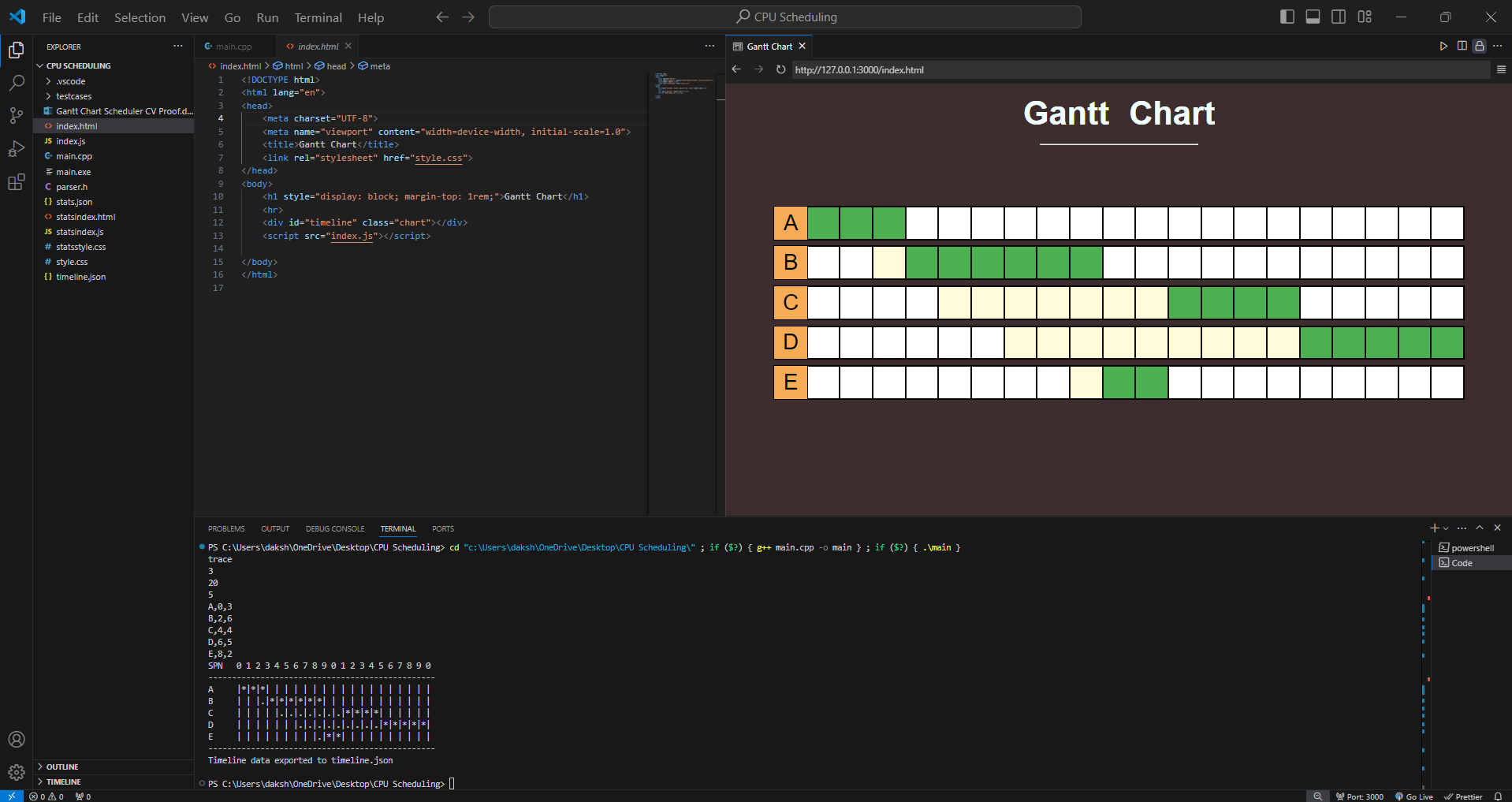
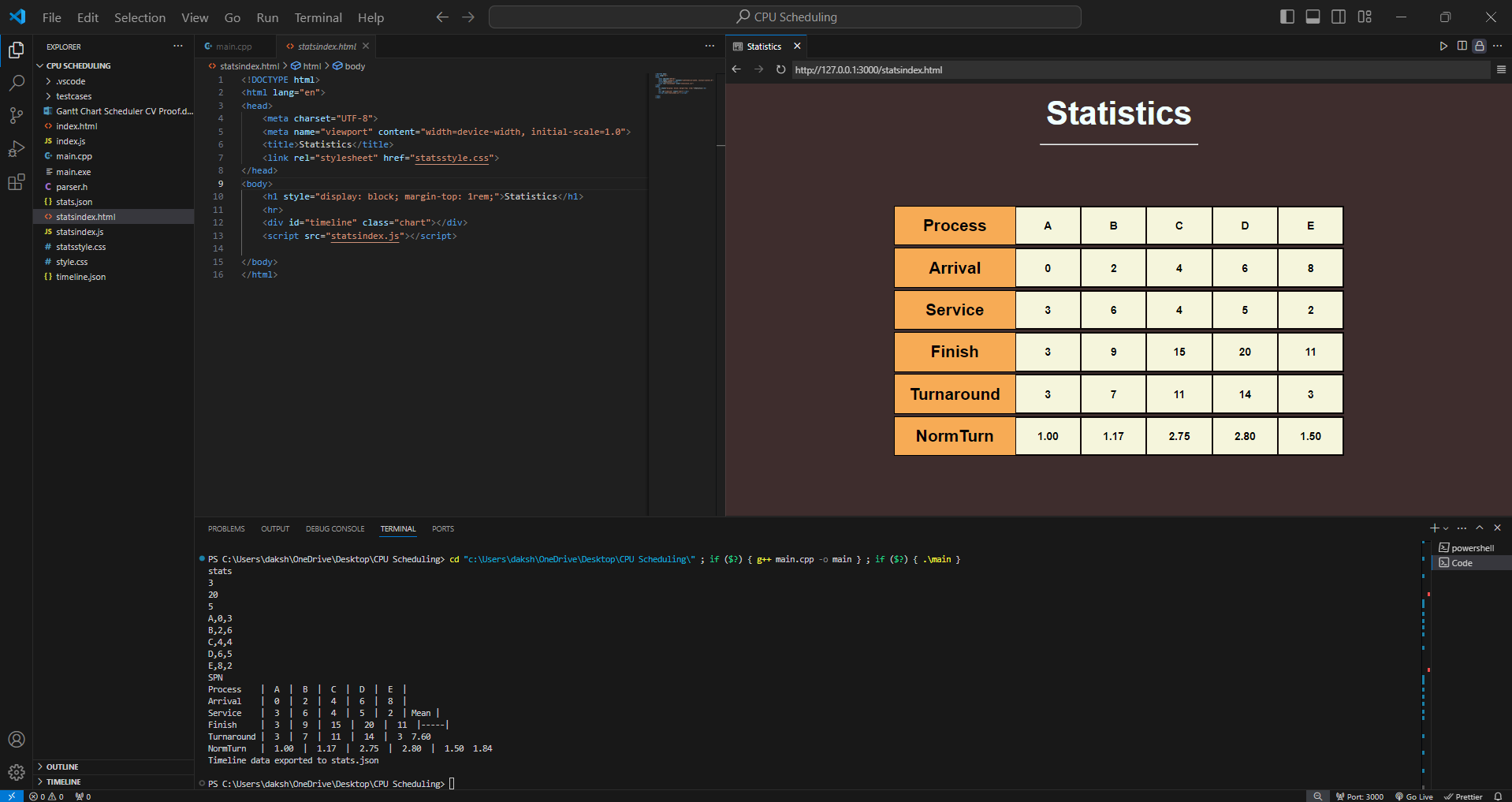


Trace and Statistics of an Example schedule 2:





Trace and Statistics of an Example schedule 3

 :