

Project Report: Leaderboard Generator and Request Logger

Introduction:

This app is a leaderboard management system that tracks player scores across multiple rounds. It updates the leaderboard in real-time and displays results in both a traditional table and a modern card view. The app logs all requests and responses for transparency and debugging. At the end, it generates an HTML file with the final leaderboard, allowing users to easily view rankings and scores.

Project Description:

The goal of this project is to create an automated system that generates clear, visually appealing leaderboards in HTML format. It displays scores from each round side by side and provides a summary table for total scores. The project also logs all requests and responses with timestamps to monitor system performance.

Steps to Run the Code:

In order to run the project, follow the below given steps:

- 1) Clone the project and change directory (cd) to the path where the project is cloned or located.
- 2) Make sure to delete the leaderboard.html and request.log file from the folder to avoid any errors while executing. The fresh output leaderboard.html and request.log file will be created after the code execution is complete.
- 3) In the terminal, run '**stack run**'. This will initialize 10 clients to send the request of their scores for 10 rounds, which will initiate the web server to sort the scores for each round in descending order.
- 4) After the execution is completed, leaderboard.html and request.log file gets created in the current directory folder. The leaderboard.html file contains the visual representation of the Score board in table format for each round, and final scores in card view format. The simulated request and response is seen in the request.log file.

Key Features of the Leaderboard System:

- 1) **Dynamic Leaderboard Generation:** The system dynamically creates HTML leaderboards for multiple competition rounds. Each round's scores are displayed

side by side, and a dedicated section summarizes the total scores across all rounds.

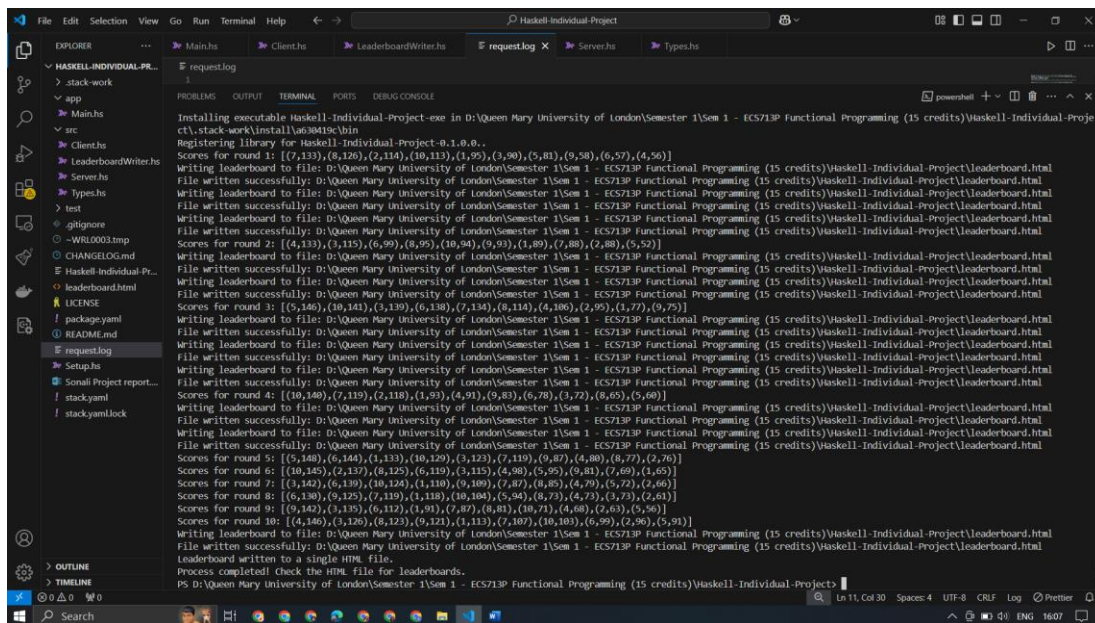
- 2) **Highlighting Winners:** The leaderboard emphasizes the winners of each round by displaying their details prominently below the corresponding tables. Additionally, it highlights the overall winner at the end of the leaderboard.
- 3) **Request and Response Logging:** All system actions are logged in a request.log file, including timestamps for each request and response. This ensures easy tracking, debugging, and transparency.
- 4) **Responsive and Customizable Design:** Designed to look great on all devices, the leaderboard features a responsive layout. It also allows for customizable styles, making the presentation visually appealing and adaptable to different needs.

Design Decisions for the Project

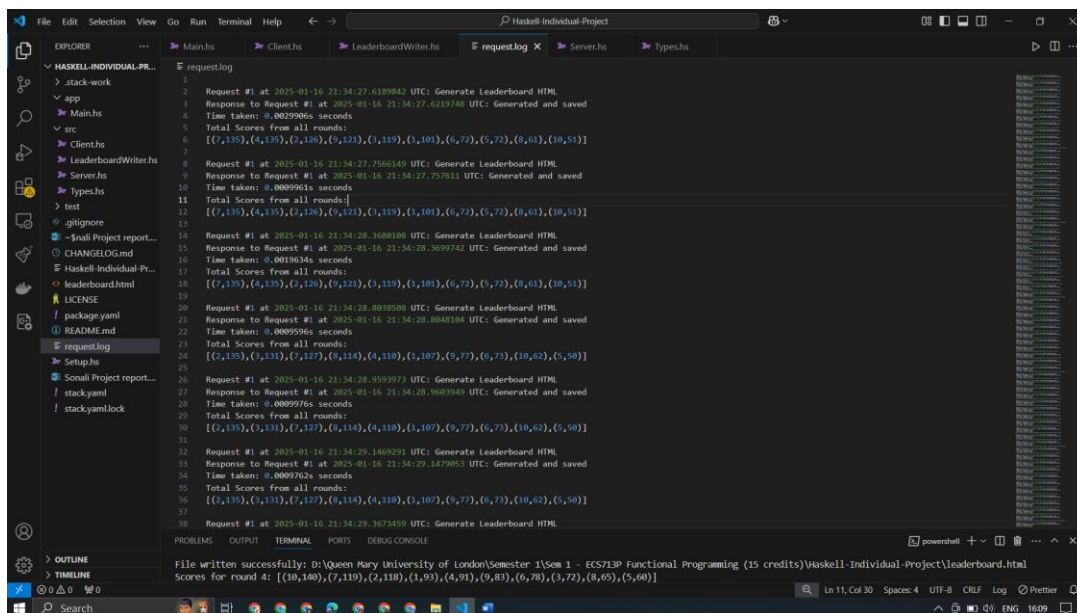
1. **Modular Approach:** The project is divided into separate modules (e.g., LeaderboardWriter, Types, and Server). This keeps the code organized, easier to maintain, and more scalable in the future.
2. **Logging Requests and Responses:** Every request and response are logged with key details such as timestamps and content. This provides transparency, helps with debugging, and allows for monitoring system behavior.
3. **Leaderboards in Multiple Views:** The leaderboard is displayed in both **Tabular** and **Card** views. This provides users with a clean, organized display while also offering a modern, visually appealing layout in the card view.
4. **Concurrency with MVar:** To ensure thread safety, shared data like the leaderboard and request queue are managed using MVar, which prevents data corruption in concurrent scenarios.
5. **Separation of Logic and HTML:** Business logic is kept separate from HTML generation, following the principle of **Separation of Concerns**. This makes the code more maintainable and allows easier updates to the presentation without affecting the logic.
6. **Responsive CSS:** The HTML is styled using Flexbox to ensure the layout adapts across different screen sizes, offering a good experience on both mobile and desktop.

- Scalability and Extensibility:** The design is built with future features in mind, like adding user avatars or more complex sorting/filtering of leaderboard data, ensuring the system can grow as needs evolve.

Output: As shown in below image, we receive the output after we do '**stack run**' in the terminal



Request.log: As shown in below image, the logs are written in request.log file



Leaderboard.html: As shown in below image, the visual representation of the request and response pairs is written to leaderboard.html file.

LEADER-BOARD

Round 1

Rank	Client ID	Total Score
1	7	133
2	8	126
3	2	114
4	10	113
5	1	95
6	3	90
7	5	81
8	9	58
9	6	57
10	4	56

Round 2

Rank	Client ID	Total Score
1	4	133
2	3	115
3	6	99
4	8	95
5	10	94
6	9	93
7	1	89
8	7	88
9	2	88
10	5	52

Round 3

Rank	Client ID	Total Score
1	5	146
2	10	141
3	3	139
4	6	138
5	7	134
6	8	114
7	4	106
8	2	95
9	1	77
10	9	75

Round 4

Rank	Client ID	Total Score
1	10	140
2	7	119
3	2	118
4	1	93
5	4	91
6	9	83
7	6	78
8	3	72
9	8	65
10	5	60

Round 5

Rank	Client ID	Total Score
1	5	148
2	6	144
3	1	133
4	10	129
5	3	123
6	7	119
7	9	87
8	4	80
9	8	77
10	2	76

Round 6

Rank	Client ID	Total Score
1	10	145
2	2	137
3	8	125

Round 7

Rank	Client ID	Total Score
1	3	142
2	6	139
3	10	124

Round 8

Rank	Client ID	Total Score
1	6	130
2	9	125
3	7	119

Round 9

Rank	Client ID	Total Score
1	9	142
2	3	135
3	6	112

Round 10

Rank	Client ID	Total Score
1	4	146
2	3	126
3	8	123

Round 6

Rank	Client ID	Total Score
1	10	145
2	2	137
3	8	125
4	6	119
5	3	115
6	4	98
7	5	95
8	9	81
9	7	69
10	1	65

Round 7

Rank	Client ID	Total Score
1	3	142
2	6	139
3	10	124
4	1	110
5	9	109
6	7	87
7	8	85
8	4	79
9	5	72
10	2	66

Round 8

Rank	Client ID	Total Score
1	6	130
2	9	125
3	7	119
4	1	118
5	10	104
6	5	94
7	8	73
8	4	73
9	3	73
10	2	61

Round 9

Rank	Client ID	Total Score
1	9	142
2	3	135
3	6	112
4	1	91
5	7	87
6	8	81
7	10	71
8	4	68
9	2	63
10	5	56

Round 10

Rank	Client ID	Total Score
1	4	146
2	3	126
3	8	123
4	9	121
5	1	113
6	7	107
7	10	103
8	6	99
9	2	96
10	5	91

TOTAL SCORES FROM ALL THE ROUNDS

Placeholder for total scores from all rounds.

5	3	115	5	9	109	5	10	104	5	7	87	5	1	113
6	4	98	6	7	87	6	5	94	6	8	81	6	7	107
7	5	95	7	8	85	7	8	73	7	10	71	7	10	103
8	9	81	8	4	79	8	4	73	8	4	68	8	6	99
9	7	69	9	5	72	9	3	73	9	2	63	9	2	96
10	1	65	10	2	66	10	2	61	10	5	56	10	5	91

TOTAL SCORES FROM ALL THE ROUNDS

Rank: 1 Client ID: 10 Score: 1164	Rank: 2 Client ID: 3 Score: 1130	Rank: 3 Client ID: 6 Score: 1115	Rank: 4 Client ID: 7 Score: 1062	Rank: 5 Client ID: 1 Score: 984	Rank: 6 Client ID: 9 Score: 974
Rank: 7 Client ID: 8 Score: 964	Rank: 8 Client ID: 4 Score: 930	Rank: 9 Client ID: 2 Score: 914	Rank: 10 Client ID: 5 Score: 895		

Conclusion: The design choices made in this project were aimed at creating a system that is modular, maintainable, and flexible. The use of separate modules, concurrency control, responsive HTML generation, and logging ensures that the system is robust and can handle future improvements. The design also balances functional requirements (accurate leaderboard processing) with user experience considerations (appealing visual presentation).