HW5 Report

Part 1: Site Design and README

Project Structure

```
CS6675_HW5/
├── server.js          # express server with routing
├── package.json       # node dependencies
└── public/            # site pages
    ├── index.html
    ├── page1.html
    ├── page2.html
    ├── page3.html
    ├── page4.html
    ├── page5.html
    └── images/
```

Features

- Hosts a very simple web server with some additional QoL packages
  - **Compression**: Uses gzip compression for better performance
  - **Logging**: Morgan middleware for HTTP request logging

Dependencies

- **express**: Web framework for Node.js
- **compression**: Gzip compression middleware
- **morgan**: HTTP request logger
- **apicache**: Caching middleware (commented out)

Installation

* This is only necessary if cloning from git, if using zip skip to **usage**

1. Clone or download the project
2. Install dependencies:

```
npm install
```

Usage

1. Start the server:

```
node server.js
```

2. Open your browser and navigate to:

```
http://localhost:3000
```

3. The server will be accessible at port 3000 and will display:

> # Welcome to My Web Server
>
> This is the main index page. Use the links below to navigate to the other pages.
>
> ## Navigation
>
> - Home
> - Page 1
> - Page 2
> - Page 3
> - Page 4
> - Page 5

   Click through the pages as needed

Server Configuration

This is easily changed in `server.js`

- **Port**: 3000
- **Host**: 0.0.0.0 (accessible from all network interfaces)

Optional Features

The server includes code for **caching** using apicache middleware. This **significantly improves performance.** To enable these features, uncomment the lines in `server.js`

For performance testing instructions see section 3

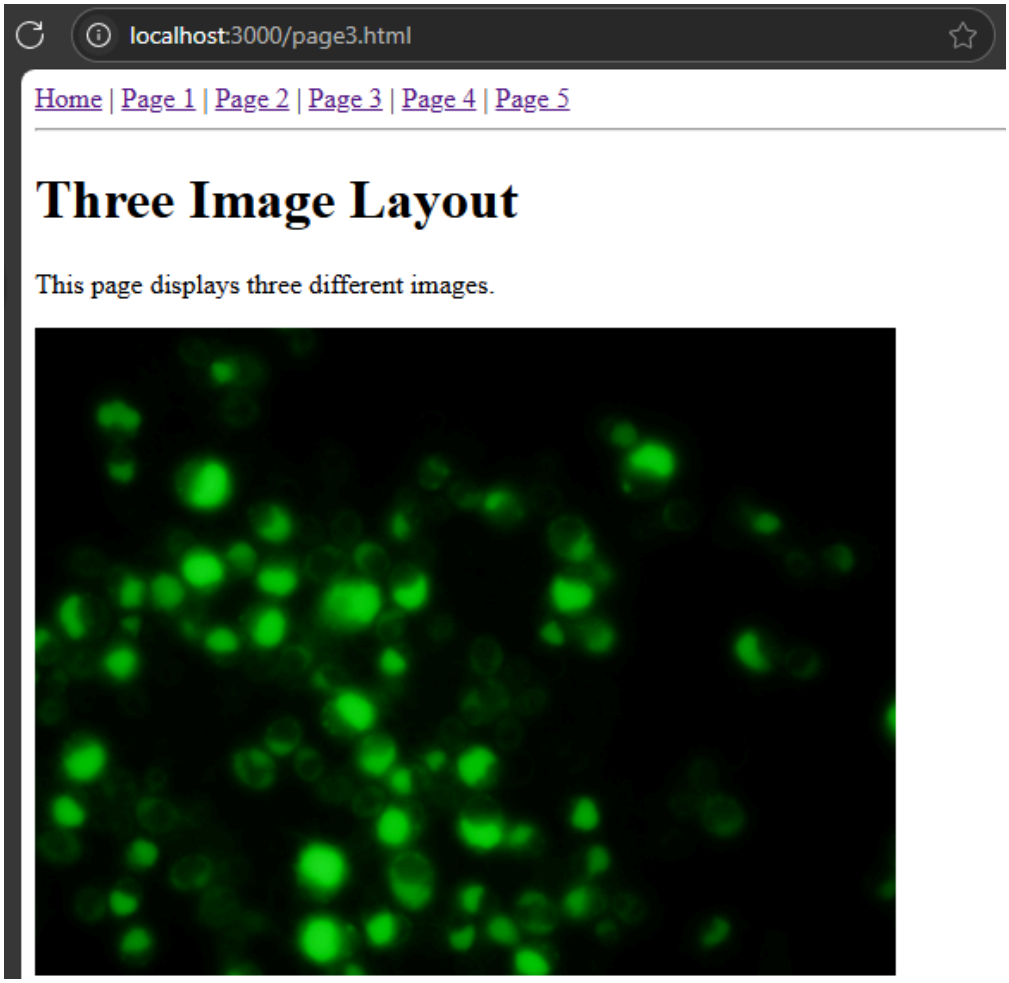
Part 2: Remote Client Access

I set up remote client access by utilizing tailscale. Tailscale is a network overlay that sits on top of any existing network infrastructure creating a private network allowing devices on the tailscale network to freely discover each other as if they were on the same local network (essentially acting like a multi-device VPN). This solution was completely new to me to set up and get running but it worked quite well allowing me to access my localhost website on any network (including mobile data as seen in the mobile screenshot) **without** using a hotspot or a

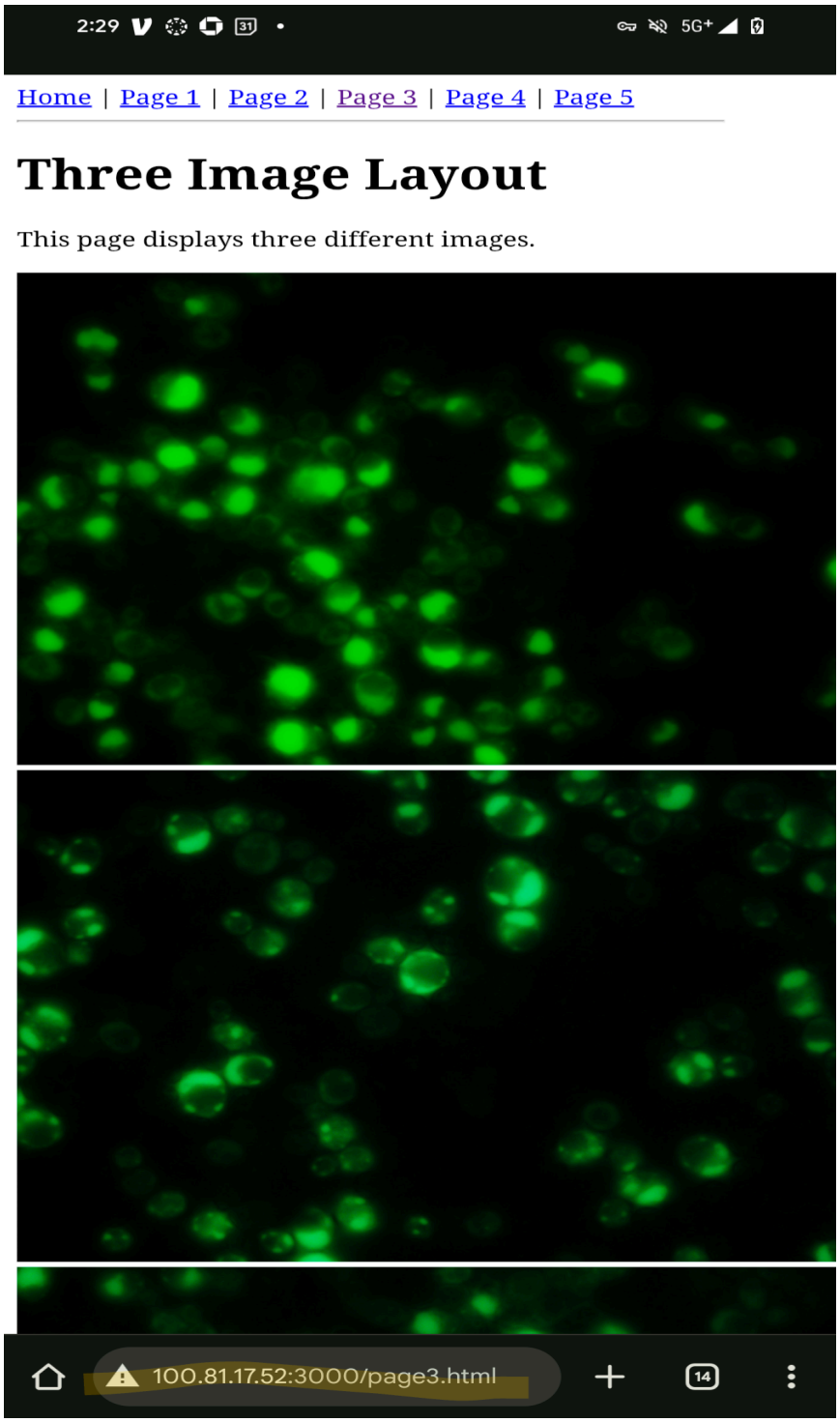
shared physical network. One positive consequence of setting up a tailscale network was that it let me freely share files across the "local" network allowing me to get these screenshots directly to my computer without any complicated sharing methods (AirDrop, cloud sync, etc.).

```
Unknown adapter Tailscale:

   Connection-specific DNS Suffix  . : tail29df4c.ts.net
   IPv6 Address. . . . . . . . . . . : fd7a:115c:a1e0::101:1134
   Link-local IPv6 Address . . . . . : fe80::51db:8521:5771:6e3b%26
   IPv4 Address. . . . . . . . . . . : 100.81.17.52
   Subnet Mask . . . . . . . . . . . : 255.255.255.255
   Default Gateway . . . . . . . . . :

Wireless LAN adapter Local Area Connection* 1:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Wireless LAN adapter Local Area Connection* 2:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Wireless LAN adapter Wi-Fi:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::a6d7:abcc:3259:da71%11
   IPv4 Address. . . . . . . . . . . : 10.146.35.107
   Subnet Mask . . . . . . . . . . . : 255.0.0.0
   Default Gateway . . . . . . . . . : 10.128.128.128

Ethernet adapter Bluetooth Network Connection:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :
```

IP-Config on Windows - See Tailscale at the top



Localhost site being hosted via Express and Node

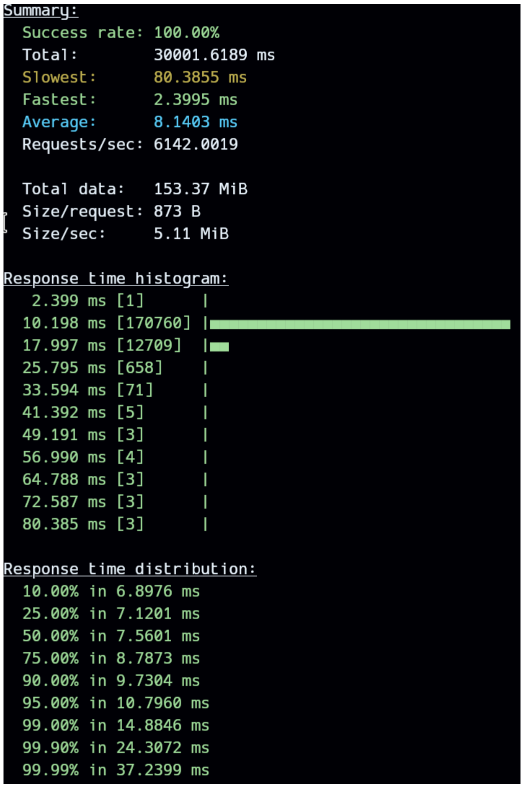


Website being accessible on my android phone

As you might notice the IP address here is different from the one in my `ipconfig` screenshot, this is because I had to switch to my Linux machine as I was running into some stability issues with windows and it had better compatibility with the benchmarking software I was using. Tailscale allowed it so that I could still link all of my devices together and get it up and running quite quickly.

Part 3 and 4: Stress Test and Latency/Throughput Analysis

The tool I used to test the performance of my web-server was called `oha`, a rust-based TUI that was modern, well-threaded, and gave great data-visibility compared to the gold-standard **apachebench**.



```
Summary:
  Success rate: 100.00%
  Total:        30001.6189 ms
  Slowest:      80.3855 ms
  Fastest:      2.3995 ms
  Average:      8.1403 ms
  Requests/sec: 6142.0019

  Total data:   153.37 MiB
  Size/request: 873 B
  Size/sec:     5.11 MiB

Response time histogram:
   2.399 ms [1]      |
  10.198 ms [170760] |■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■
  17.997 ms [12709]  |■■
  25.795 ms [658]    |
  33.594 ms [71]     |
  41.392 ms [5]      |
  49.191 ms [3]      |
  56.990 ms [4]      |
  64.788 ms [3]      |
  72.587 ms [3]      |
  80.385 ms [3]      |

Response time distribution:
  10.00% in 6.8976 ms
  25.00% in 7.1201 ms
  50.00% in 7.5601 ms
  75.00% in 8.7873 ms
  90.00% in 9.7304 ms
  95.00% in 10.7960 ms
  99.00% in 14.8846 ms
  99.90% in 24.3072 ms
  99.99% in 37.2399 ms
```

```
Progress
                                        4s / 30s

Stats for last sec                          Status code distribution
Requests : 3579                             [200] 15701 responses
Slowest: 0.1592 secs
Fastest: 0.0979 secs
Average: 0.1130 secs
Data: 2.98 MiB
Number of open files: 413 / 1024

Error distribution
```

Some examples of the outputs of `oha`, whereas apachebench simply outputs a couple of text lines of data.

The command used to run the following tests was as follows:
`oha -z s -c r http://localhost:3000/page1.html`
where

- -z s is how many seconds to run the test (I used 30 seconds)
  - This can be replaced by -n for number of requests
- -c r is how many concurrent requests per second (this was modified)

Performance Results (No Caching)

Test Run	Concurrent Users	Requests/sec (Throughput)	Avg Latency (ms)	Fastest (ms)	Slowest (ms)
Latency Test	10	5,932	1.69	0.85	16.20
Latency Test	50	**6,142**	8.14	2.40	80.39
Stress Test	200	5,935	33.71	20.02	721.40
Stress Test (Re-run)	400	5,906	67.70	45.00	2,588.70
Stress Test	1000	5,527	181.00	47.00	1,703.40

Without caching, the server handled a max of 6,142 req/sec, with it slowly leveling off as concurrent users grew, showing that the server slowly became overloaded. We also see this with the dramatic jump in latency as we 100x users our latency grows accordingly. Looking at the Response Time Percentiles chart as well we see that the range in response times grows quite significantly.

Performance Results (Caching)

Test Run	Concurrent Users	Requests/sec (Throughput)	Avg Latency (ms)	Fastest (ms)	Slowest (ms)
Latency Test	10	**8,401**	1.20	0.10	24.73
Latency Test	50	8,250	6.06	0.15	246.89
Stress Test	200	7,868	25.40	4.60	2,044.20
Stress Test	400	7,850	50.90	1.50	5,787.10
Stress Test	1000	7,955	106.00	26.00	2,984.00

With caching, the server stores copies of pages and is able to serve it instantly instead of rebuilding on every request. With caching we see a significant jump in throughout (an almost ~37% increase when comparing peaks). While the pattern for throughput and latency is similar to our data without caching (suggesting that caching gets to a point of diminishing returns), we see that the average latency across the board is lower (~41% decrease when comparing worst latency). This shows that caching is a resounding success especially when we look at the median response times which show an even greater improvement than when looking at averages that take into account the large outliers.

Response Time Percentiles (ms)

Test Run	Concurrent Users	P50 (Median)	P75	P90	P95	P99
Latency Test	10	1.51	1.61	2.01	2.99	3.92
Latency Test	50	7.56	8.79	9.73	10.80	14.88
Stress Test	200	32.67	34.81	37.52	40.60	50.78
Stress Test (Re-run)	400	65.50	68.50	74.50	79.30	90.20
Stress Test	1000	170.00	180.00	191.00	199.00	206.00

Response Time Percentiles (ms) - Caching

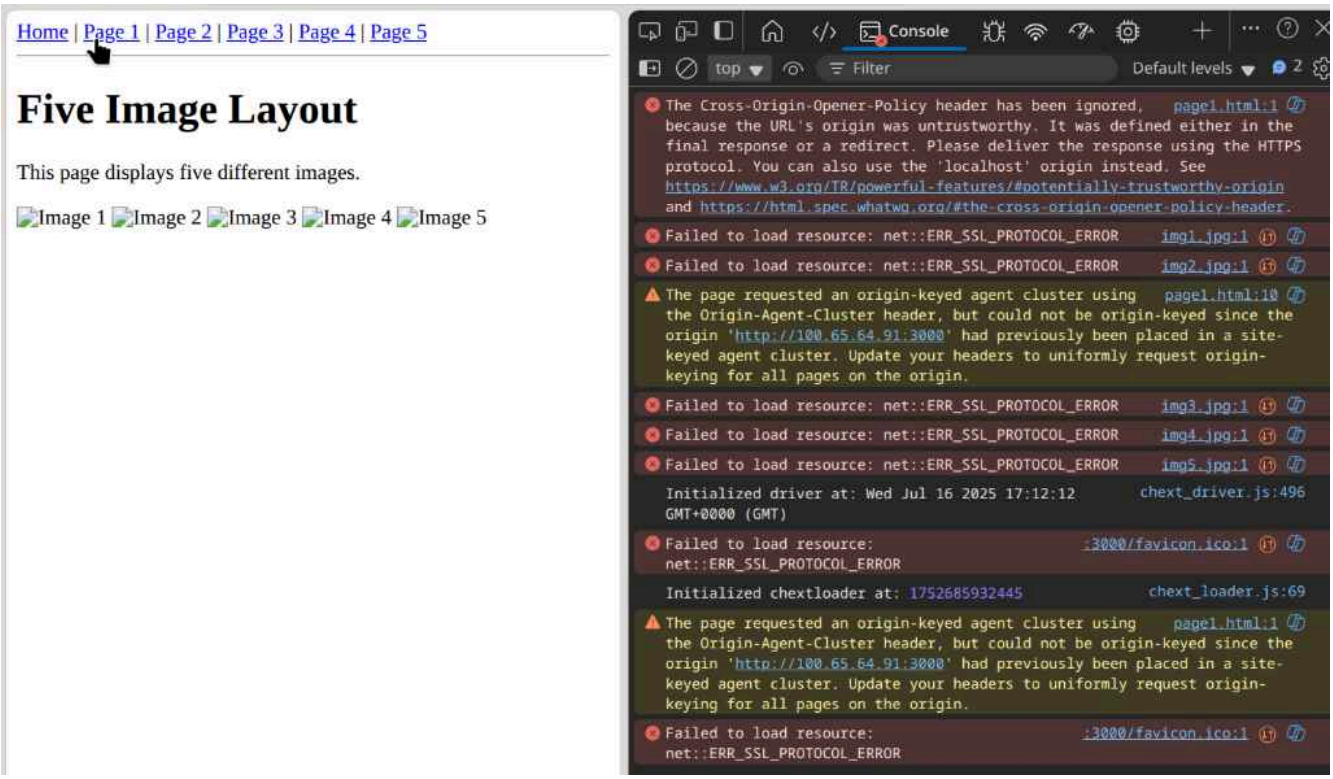
Test Run	Concurrent Users	P50 (Median)	P75	P90	P95	P99
Latency Test	10	1.03	1.09	1.40	2.40	3.35
Latency Test	50	5.40	6.76	7.71	8.65	11.53
Stress Test	200	24.00	25.80	30.10	32.70	38.40
Stress Test	400	48.20	50.90	56.20	61.60	78.90
Stress Test	1000	85.00	96.00	103.00	109.00	123.00

Experience

Overall, I thought this was a really cool project, as I pushed myself to learn a variety of new toolkits and software. Coming into this I had some familiarity with Apache TomCat so I wanted to explore newer more modern frameworks. I decided on using `Express.js` and Node.js because both are relatively modern and they are becoming increasingly popular amongst web developers today, it is well documented and very easy and quick to get up and running. In addition, it is very flexible allowing me to work with and explore different packages (like morgan and compression), which provided me with greater logging visibility and improved web-server performance right off the bat. Building the web server itself is where I ran into a couple of learnings:

1. Browser security policies have evolved a lot. Setting up the server I ran into numerous issues with multiple different browsers forcing my site to run with HTTPS even if the main site loaded in via HTTP. This is where I learned about HSTS or HTTP Strict Transport Security which tells browsers to always use HTTPS which while **not** configured for localhost was set to automatically work on my tailscale IP addresses which lead to quite a headache trying to figure out why my http requests where being sent as HTTPS and promptly

rejected.



2. The second thing I had to learn was how to actually access my webserver. Normally I would just find the IP of the device hosting the site and connect to it (say at my home router), but this failed to work in my apartment and on campus leading to a lot of confusion as to why. After doing some digging I found that this was a security feature that leads to device discover-ability being disabled. I searched for solutions like setting up a hotspot, which worked but ate up a lot of my bandwidth as I made the images quite large on my website. I also tried ZeroTier which allowed for the creation of a private network but was fairly complicated so eventually I settled on tailscale which had a very intuitive quick start guide and allowed me to get a private network up and running quite quickly.

In the theme of using modern tools, I wanted to use a different tool than apachebench which came preinstalled with my fedora install and is well-known for being quick and easy to use. The issue was that the application is famously singlethreaded which could lead to performance issues during benchmarking, so I looked for packages built on a more modern framework and came across `oha` which is a rust based benchmarking tool that had an excellent TUI that gave quite a bit of insight on what was happening at a network level and how the requests were being handled on my webserver. This combined with the histograms for request data made analysis of the data easier than if I was to use apachebench. Which leads me to learning 3:
3. Logging is incredibly helpful, for example in the first learning about security. Web-server logging is quite complete and thorough from the server side of things to the browser side of things with developer tools and being able to inspect network queries. Developer tools combined with the logging package I was using (Morgan), allowed me to identify that a security package I was using called helmet was erroneously denying mass requests leading to performance numbers that didn't make sense at all. Eventually I realized that the requests were being denied and I have since removed the package. Logging also helped with `oha` as it allowed me to see exactly what was happening when I ran the stress test to make sure that requests were being properly handled, I ran into an issue where the cached requests were seemingly performing worse than the uncached requests and I narrowed it down to how long it would take for my cache to dump stale entries (it was far too short).

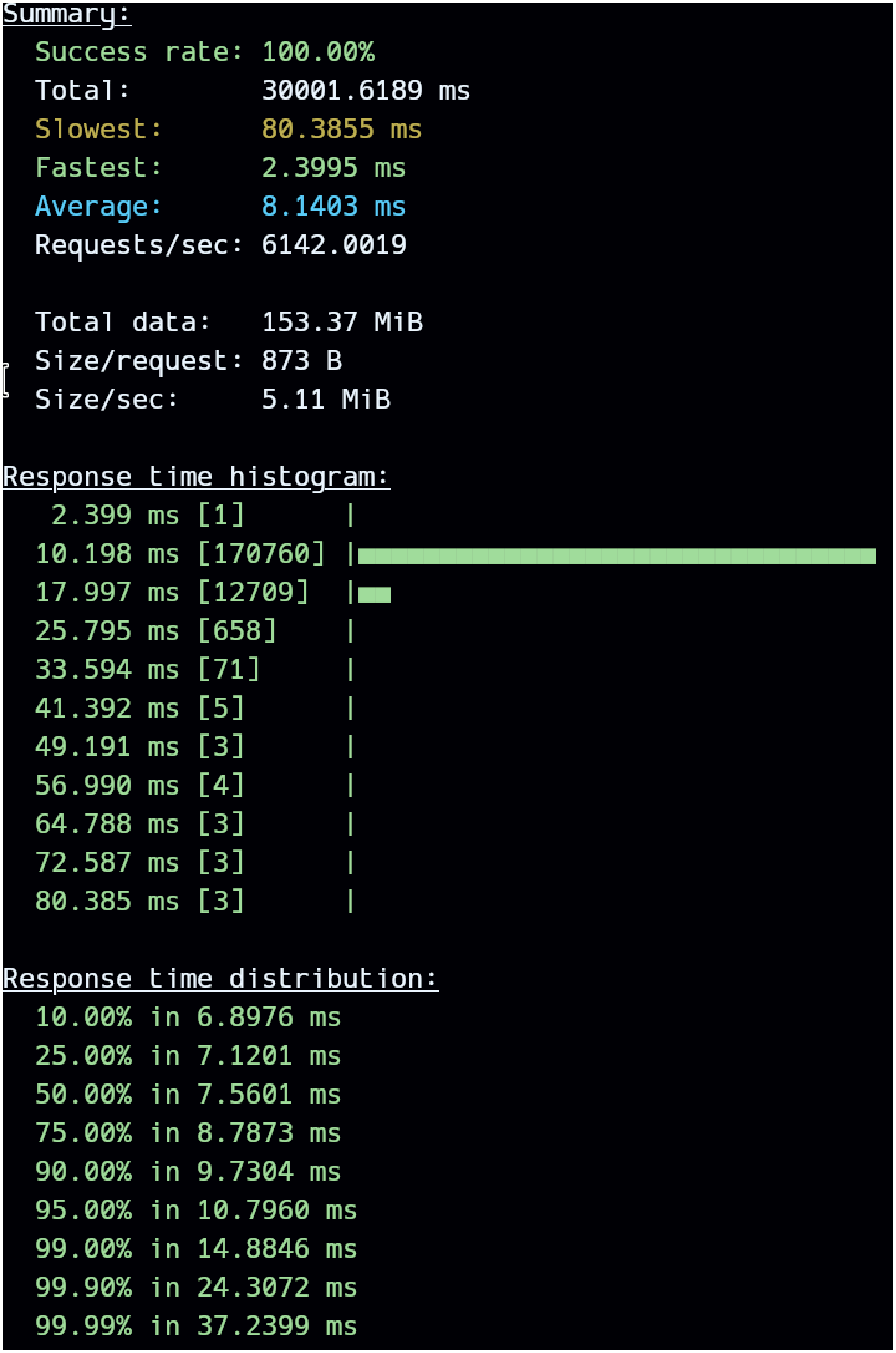
Appendix (Raw Results)

Part 3 without Caching

`oha -z 30s -c 50 http://localhost:3000/page1.html` (Without Caching)

```
Summary:
  Success rate: 100.00%
  Total:    30001.6189 ms
  Slowest:  80.3855 ms
  Fastest:  2.3995 ms
  Average:  8.1403 ms
  Requests/sec: 6142.0019

  Total data:   153.37 MiB
  Size/request: 873 B
  Size/sec: 5.11 MiB

Response time histogram:
   2.399 ms [1]      |
  10.198 ms [170760] |■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■
  17.997 ms [12709]  |■■
  25.795 ms [658]    |
  33.594 ms [71]     |
  41.392 ms [5]      |
  49.191 ms [3]      |
  56.990 ms [4]      |
  64.788 ms [3]      |
  72.587 ms [3]      |
  80.385 ms [3]      |

Response time distribution:
  10.00% in 6.8976 ms
  25.00% in 7.1201 ms
  50.00% in 7.5601 ms
  75.00% in 8.7873 ms
  90.00% in 9.7304 ms
  95.00% in 10.7960 ms
  99.00% in 14.8846 ms
  99.90% in 24.3072 ms
  99.99% in 37.2399 ms
```

Visual Representation of the TUI



```
Summary:
  Success rate: 100.00%
  Total:        30001.6189 ms
  Slowest:      80.3855 ms
  Fastest:      2.3995 ms
  Average:      8.1403 ms
  Requests/sec: 6142.0019

  Total data:   153.37 MiB
  Size/request: 873 B
  Size/sec:     5.11 MiB

Response time histogram:
   2.399 ms [1]      |
  10.198 ms [170760] |■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■
  17.997 ms [12709]  |■■
  25.795 ms [658]    |
  33.594 ms [71]     |
  41.392 ms [5]      |
  49.191 ms [3]      |
  56.990 ms [4]      |
  64.788 ms [3]      |
  72.587 ms [3]      |
  80.385 ms [3]      |

Response time distribution:
  10.00% in 6.8976 ms
  25.00% in 7.1201 ms
  50.00% in 7.5601 ms
  75.00% in 8.7873 ms
  90.00% in 9.7304 ms
  95.00% in 10.7960 ms
  99.00% in 14.8846 ms
  99.90% in 24.3072 ms
  99.99% in 37.2399 ms
```

`oha -z 30s -c 10 http://localhost:3000/page1.html`

```
Summary:
  Success rate: 100.00%
  Total:    30001.1710 ms
  Slowest:  16.1964 ms
  Fastest:  0.8513 ms
  Average:  1.6850 ms
  Requests/sec: 5932.1018

  Total data:   148.16 MiB
  Size/request: 873 B
  Size/sec: 4.94 MiB

Response time histogram:
   0.851 ms [1]      |
   2.386 ms [162761] |■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■
   3.920 ms [13423]  |■■
   5.455 ms [1382]   |
   6.989 ms [297]    |
   8.524 ms [67]     |
  10.058 ms [19]     |
  11.593 ms [4]      |
  13.127 ms [1]      |
  14.662 ms [1]      |
  16.196 ms [5]      |

Response time distribution:
  10.00% in 1.4333 ms
  25.00% in 1.4679 ms
  50.00% in 1.5116 ms
  75.00% in 1.6109 ms
  90.00% in 2.0064 ms
  95.00% in 2.9872 ms
  99.00% in 3.9197 ms
  99.90% in 6.1850 ms
  99.99% in 9.1293 ms


Details (average, fastest, slowest):
  DNS+dialup:   0.4255 ms, 0.2872 ms, 0.5384 ms
  DNS-lookup:   0.1721 ms, 0.0070 ms, 0.3988 ms

Status code distribution:
  [200] 177961 responses
```

Part 4 - Stress Test (No caching)

`oha -z 30s -c 200 http://localhost:3000/page1.html`

```
Summary:
  Success rate: 100.00%
  Total:    30002.9344 ms
  Slowest:  721.4011 ms
```

```
  Fastest:  20.0199 ms
  Average:  33.7097 ms
  Requests/sec: 5934.5862

  Total data:   148.08 MiB
  Size/request: 873 B
  Size/sec: 4.94 MiB

Response time histogram:
   20.020 ms [1]      |
   90.158 ms [177773] |■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■
  160.296 ms [10]     |
  230.434 ms [11]     |
  300.572 ms [13]     |
  370.711 ms [10]     |
  440.849 ms [11]     |
  510.987 ms [9]      |
  581.125 ms [10]     |
  651.263 ms [9]      |
  721.401 ms [8]      |

Response time distribution:
  10.00% in 29.8013 ms
  25.00% in 31.2252 ms
  50.00% in 32.6732 ms
  75.00% in 34.8140 ms
  90.00% in 37.5157 ms
  95.00% in 40.5963 ms
  99.00% in 50.7841 ms
  99.90% in 77.5339 ms
  99.99% in 579.9252 ms


Details (average, fastest, slowest):
  DNS+dialup:   12.3355 ms, 0.1685 ms, 21.7581 ms
  DNS-lookup:   0.0269 ms, 0.0012 ms, 0.7919 ms

Status code distribution:
  [200] 177865 responses
```

`oha -z 30s -c 400 http://localhost:3000/page1.html`

```
Summary:
  Success rate: 100.00%
  Total:    30.0057 sec
  Slowest:  2.0149 sec
  Fastest:  0.0465 sec
  Average:  0.0697 sec
  Requests/sec: 5747.1054

  Total data:   143.24 MiB
  Size/request: 873 B
  Size/sec: 4.77 MiB
```

```
Response time histogram:
  0.046 sec [1]      |
  0.243 sec [171922] |■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■
  0.440 sec [16]     |
  0.637 sec [15]     |
  0.834 sec [15]     |
  1.031 sec [14]     |
  1.228 sec [14]     |
  1.424 sec [14]     |
  1.621 sec [12]     |
  1.818 sec [12]     |
  2.015 sec [12]     |

Response time distribution:
  10.00% in 0.0629 sec
  25.00% in 0.0654 sec
  50.00% in 0.0680 sec
  75.00% in 0.0717 sec
  90.00% in 0.0770 sec
  95.00% in 0.0809 sec
  99.00% in 0.0912 sec
  99.90% in 0.1310 sec
  99.99% in 1.7349 sec


Details (average, fastest, slowest):
  DNS+dialup:   0.0129 sec, 0.0003 sec, 0.0371 sec
  DNS-lookup:   0.0000 sec, 0.0000 sec, 0.0006 sec

Status code distribution:
  [200] 172047 responses
```

`oha -z 30s -c 1000 http://localhost:3000/page1.html`

```
Summary:
  Success rate: 100.00%
  Total:    3.0011 10 sec
  Slowest:  1.7034 10 sec
  Fastest:  0.0047 10 sec
  Average:  0.0181 10 sec
  Requests/sec: 5527.1694

  Total data:   137.28 MiB
  Size/request: 873 B
  Size/sec: 4.57 MiB

Response time histogram:
  0.005 10 sec [1]      |
  0.175 10 sec [164459] |■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■
  0.344 10 sec [63]     |
  0.514 10 sec [56]     |
  0.684 10 sec [53]     |
```

```
  0.854 10 sec [47]     |
  1.024 10 sec [48]     |
  1.194 10 sec [42]     |
  1.364 10 sec [40]     |
  1.534 10 sec [40]     |
  1.703 10 sec [39]     |

Response time distribution:
  10.00% in 0.0109 10 sec
  25.00% in 0.0137 10 sec
  50.00% in 0.0170 10 sec
  75.00% in 0.0180 10 sec
  90.00% in 0.0191 10 sec
  95.00% in 0.0199 10 sec
  99.00% in 0.0206 10 sec
  99.90% in 1.0115 10 sec
  99.99% in 1.6330 10 sec


Details (average, fastest, slowest):
  DNS+dialup:   0.0012 10 sec, 0.0000 10 sec, 0.0047 10 sec
  DNS-lookup:   0.0000 10 sec, 0.0000 10 sec, 0.0000 10 sec

Status code distribution:
  [200] 164888 responses
```

Part 5 - Latency and Throughput (Caching)

`oha -z 30s -c 10 http://localhost:3000/page1.html`

```
Summary:
  Success rate: 100.00%
  Total:    30000.4989 ms
  Slowest:  24.7277 ms
  Fastest:  0.1005 ms
  Average:  1.1897 ms
  Requests/sec: 8400.7936

  Total data:   209.82 MiB
  Size/request: 873 B
  Size/sec: 6.99 MiB

Response time histogram:
   0.100 ms [1]      |
   2.563 ms [242438] |■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■
   5.026 ms [8801]   |■
   7.489 ms [630]    |
   9.951 ms [140]    |
  12.414 ms [4]      |
  14.877 ms [1]      |
  17.340 ms [1]      |
  19.802 ms [1]      |
  22.265 ms [1]      |
```

```
  24.728 ms [1]      |

Response time distribution:
  10.00% in 0.9933 ms
  25.00% in 1.0068 ms
  50.00% in 1.0333 ms
  75.00% in 1.0904 ms
  90.00% in 1.4034 ms
  95.00% in 2.3994 ms
  99.00% in 3.3475 ms
  99.90% in 6.8750 ms
  99.99% in 8.7983 ms


Details (average, fastest, slowest):
  DNS+dialup:   0.4469 ms, 0.1914 ms, 0.6863 ms
  DNS-lookup:   0.2882 ms, 0.0558 ms, 0.5218 ms

Status code distribution:
  [200] 252019 responses
```

```
oha -z 30s -c 50 http://localhost:3000/page1.html
```

```
Summary:
  Success rate: 100.00%
  Total:    30000.8902 ms
  Slowest:  246.8897 ms
  Fastest:  0.1493 ms
  Average:  6.0600 ms
  Requests/sec: 8250.2219

  Total data:   206.03 MiB
  Size/request: 873 B
  Size/sec: 6.87 MiB

Response time histogram:
    0.149 ms [1]      |
   24.823 ms [247421] |■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■
   49.497 ms [11]     |
   74.171 ms [5]      |
   98.845 ms [5]      |
  123.519 ms [5]      |
  148.194 ms [3]      |
  172.868 ms [4]      |
  197.542 ms [3]      |
  222.216 ms [4]      |
  246.890 ms [4]      |

Response time distribution:
  10.00% in 5.0942 ms
  25.00% in 5.2001 ms
  50.00% in 5.3954 ms
  75.00% in 6.7622 ms
```

```
  90.00% in 7.7139 ms
  95.00% in 8.6535 ms
  99.00% in 11.5281 ms
  99.90% in 19.4486 ms
  99.99% in 91.2082 ms


Details (average, fastest, slowest):
  DNS+dialup:   0.2971 ms, 0.0961 ms, 0.7209 ms
  DNS-lookup:   0.0733 ms, 0.0012 ms, 0.5982 ms

Status code distribution:
  [200] 247466 responses

Error distribution:
  [48] aborted due to deadline
```

Part 6 - Stress Test (caching)

`oha -z 30s -c 200 http://localhost:3000/page1.html`

```
Summary:
  Success rate: 100.00%
  Total:    30.0028 sec
  Slowest:  2.0442 sec
  Fastest:  0.0046 sec
  Average:  0.0254 sec
  Requests/sec: 7867.7265

  Total data:   196.37 MiB
  Size/request: 873 B
  Size/sec: 6.55 MiB

Response time histogram:
  0.005 sec [1]      |
  0.209 sec [235767] |■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■
  0.412 sec [12]     |
  0.616 sec [14]     |
  0.820 sec [10]     |
  1.024 sec [11]     |
  1.228 sec [10]     |
  1.432 sec [10]     |
  1.636 sec [9]      |
  1.840 sec [9]      |
  2.044 sec [9]      |

Response time distribution:
  10.00% in 0.0221 sec
  25.00% in 0.0232 sec
  50.00% in 0.0240 sec
  75.00% in 0.0258 sec
  90.00% in 0.0301 sec
  95.00% in 0.0327 sec
```

```
  99.00% in 0.0384 sec
  99.90% in 0.0542 sec
  99.99% in 1
```

`oha -z 30s -c 400 http://localhost:3000/page1.html`

```
Summary:
  Success rate: 100.00%
  Total:    30.0042 sec
  Slowest:  5.7871 sec
  Fastest:  0.0015 sec
  Average:  0.0509 sec
  Requests/sec: 7850.4754

  Total data:   195.78 MiB
  Size/request: 873 B
  Size/sec: 6.53 MiB

Response time histogram:
  0.002 sec [1]      |
  0.580 sec [235036] |■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■
  1.159 sec [15]     |
  1.737 sec [14]     |
  2.316 sec [13]     |
  2.894 sec [14]     |
  3.473 sec [12]     |
  4.051 sec [13]     |
  4.630 sec [13]     |
  5.209 sec [12]     |
  5.787 sec [12]     |

Response time distribution:
  10.00% in 0.0436 sec
  25.00% in 0.0464 sec
  50.00% in 0.0482 sec
  75.00% in 0.0509 sec
  90.00% in 0.0562 sec
  95.00% in 0.0616 sec
  99.00% in 0.0789 sec
  99.90% in 0.1146 sec
  99.99% in 4.6471 sec


Details (average, fastest, slowest):
  DNS+dialup:   0.0165 sec, 0.0003 sec, 0.0438 sec
  DNS-lookup:   0.0000 sec, 0.0000 sec, 0.0018 sec

Status code distribution:
  [200] 235155 responses
```

`oha -z 30s -c 1000 http://localhost:3000/page1.html`

```
Summary:
  Success rate: 100.00%
  Total:    3.0014 10 sec
  Slowest:  2.9840 10 sec
  Fastest:  0.0026 10 sec
  Average:  0.0106 10 sec
  Requests/sec: 7955.3330

  Total data:   197.98 MiB
  Size/request: 873 B
  Size/sec: 6.60 MiB


Response time histogram:
  0.003 10 sec [1]      |
  0.301 10 sec [237487] |■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■
  0.599 10 sec [43]     |
  0.897 10 sec [39]     |
  1.195 10 sec [38]     |
  1.493 10 sec [37]     |
  1.791 10 sec [34]     |
  2.090 10 sec [32]     |
  2.388 10 sec [31]     |
  2.686 10 sec [30]     |
  2.984 10 sec [29]     |

Response time distribution:
  10.00% in 0.0066 10 sec
  25.00% in 0.0075 10 sec
  50.00% in 0.0085 10 sec
  75.00% in 0.0096 10 sec
  90.00% in 0.0103 10 sec
  95.00% in 0.0109 10 sec
  99.00% in 0.0123 10 sec
  99.90% in 0.8439 10 sec
  99.99% in 2.7410 10 sec


Details (average, fastest, slowest):
  DNS+dialup:   0.0018 10 sec, 0.0000 10 sec, 0.0054 10 sec
  DNS-lookup:   0.0000 10 sec, 0.0000 10 sec, 0.0001 10 sec

Status code distribution:
  [200] 237801 responses
```

No Cache Stress test re-run

```
oha -z 30s -c 400 http://localhost:3000/page1.html
Summary:
  Success rate: 100.00%
  Total:    30.0045 sec
  Slowest:  2.5887 sec
```

```
    Fastest:  0.0450 sec
    Average:  0.0677 sec
    Requests/sec: 5906.3446

    Total data:   147.22 MiB
    Size/request: 873 B
    Size/sec: 4.91 MiB

  Response time histogram:
    0.045 sec [1]      |
    0.299 sec [176671] |■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■
    0.554 sec [23]     |
    0.808 sec [17]     |
    1.062 sec [18]     |
    1.317 sec [20]     |
    1.571 sec [18]     |
    1.826 sec [16]     |
    2.080 sec [16]     |
    2.334 sec [16]     |
    2.589 sec [15]     |

  Response time distribution:
    10.00% in 0.0608 sec
    25.00% in 0.0629 sec
    50.00% in 0.0655 sec
    75.00% in 0.0685 sec
    90.00% in 0.0745 sec
    95.00% in 0.0793 sec
    99.00% in 0.0902 sec
    99.90% in 0.1336 sec
    99.99% in 2.2871 sec


  Details (average, fastest, slowest):
    DNS+dialup:   0.0193 sec, 0.0003 sec, 0.0456 sec
    DNS-lookup:   0.0000 sec, 0.0000 sec, 0.0008 sec

  Status code distribution:
    [200] 176831 responses
```