National College of Ireland

<BSHCYB4>

<Cybersecurity>

<Academic Year i.e. 2024/2025>

<Price Asemota>

<x21445372>

<x21445372@student.ncirl.ie>

<P-VPN method>

Technical Report

Contents

Executive Summary

Max 300 words. Summarise the key points of the report. Restate the purpose of the report, highlight the major points of the report, and describe any results, conclusions, or recommendations from the report.

This project report describes VPN protocol evaluation with development progress to demonstrate my work as well as keeping documentation of constructing the application independently. Investing time to review this work shows my technical advancement along

with my obtained achievements during development. This project analyses the performance aspects of Proton VPN, OpenVPN and WireGuard VPN protocols through examinations of speed, traffic regulation, processing resource usage and total cost effectiveness.

Protecting user privacy through trusted VPN services remains the central reason for this study since it focuses on identifying the boundaries of insecure public Wi-Fi access and explaining adequate security tools. The project uses a custom-built web application with speed test API to generate instant detection data which improves both usefulness and credibility of the comparison process.

A part of this project involves keeping past test results while the interface provides users with clear performance metrics under standardized conditions.

This report delivers both technological VPN solution comparisons and establishes a guide system for anyone wishing to protect their internet access through secure choice selection. Technical difficulties and accomplishments from development work are represented in this summary.

1.0 Introduction

1.1. Background

Why did you undertake this project?

The reason I am doing this project is because there was a time when I was in one of my networking modules in 3rd year, my lecturer demonstrated to us how easy it was for him to access our IP addresses once we were connected to his mobile hotspot or even the public Wi-Fi we use in the college. Once seeing that I started researching VPN's I could use to make myself feel less vulnerable and from there I had the idea why not create something to show people the differences between two VPN's so I could save themselves the time to research and they could easily know what they want.

1.2. Aims

What does the project aim to achieve?

Something I would love to do is to raise awareness when it comes to using VPNs, so in this project I am going to analyse ProtonVPN, OpenVPN and Wire Guard by assessing their performance, costs, and resource consumption on cloud servers, ultimately identifying the optimal choice for securing public Wi-Fi. Additionally, it will investigate the possibility of leveraging machine learning to enhance VPN selection. I would like to make it easier for people to choose what VPN they would like to use from my research.

1.3. Technology

What technology will you use to achieve what you have set out to do and how will you use it?

I added a mixture of technologies which proved suitable for VPN performance assessments along with secure web development requirements. I tested VPN services through OpenVPN, WireGuard and ProtonVPN installations along with GUI interface and command-line connection and testing approaches under standardized test conditions. The tests ran these VPNs in different public Wi-Fi networks along with mobile hotspots to measure latency performance while monitoring download and upload speeds while monitoring system power use.

My web application incorporates a real-time speed test API which enables users to check their network speeds across different VPNs through the system.

Five paramount cybersecurity tests were performed by me to support project analysis: Packet Sniffing together with DNS Leak Testing and IP Leak Testing in addition to Speed Testing and Traffic Analysis. and Wireshark collected and examined network traffic to indicate both the threats public networks pose and the necessity of VPN security.

The application uses HTML for its development while CSS and JavaScript form its core structure together with localStorage functionality to save and show past test results. The analysis of unsecured network risks included using Wireshark and for traffic inspection. The selection of these technologies accomplished both functional performance evaluation and user-friendly design along with emphasizing VPN security requirements in risky network conditions.

1.4. Structure

Provide a brief overview of the structure of the document and what is addressed in each section.

Part 1

I am just giving a summary of my report and talking about the basis of my project and what I am doing.

Part 2:

Here I am showing use cases and explaining how it works when the user is using my webapp and testing their connection. I am also showing key parts of my code and website and showing how I implemented certain features.

Part 3:

In my conclusion I am giving some advantages and some disadvantages about my project.

Part 4:

Here I am giving some ideas as to what else I could implement into my project if I had more time to work on it.

Part 5:

I linked some websites here as reference that helped me get to where im at with my project as of this this current stage.

Part 6:

2.0 System

2.1. Requirements

All requirements should be verifiable. For example, experienced controllers shall be able to use all the system functions after a total of two hours training. After this training, the average number of errors made by experienced users shall not exceed two per day.

Usability Requirement

The web application enables users with technical understanding to achieve VPN speed test results after opening it for no more than 5 minutes. The speed test application needs to operate easily with quick response times for users attempting tests from both computers and mobile devices.

Training Requirement

Users need only 10 minutes for self-exploration or guidance to learn about the entire system functionality. The system does not need any specialized formal training program.

Performance Requirement

The speed tester measures speed along with latency data and upload and download speeds using a system response time of 10 seconds from the test start.

Reliability Requirement

The system needs to present accurate historical speed test results throughout 95% of user sessions with no data disappearance.

Security Testing Requirement

The system includes a capability to check five cybersecurity parameters with documentation features for Packet Sniffing, IP Leak, DNS Leak, Traffic Analysis and Speed Testing. The tests must output definitive results which duplicate tests performed under various VPN configurations.

Compatibility Requirement

The system needs to operate appropriately when utilizing OpenVPN, WireGuard and ProtonVPN regardless of the connection type including public Wi-Fi and mobile hotspot.

Error Rate Requirement

After initial use users must not encounter more than two errors which include failed speed tests and broken table displays during each session.

2.1.1. Functional Requirements

This section lists the functional requirements in **ranked order**. Functional requirements describe the possible effects of a software system, in other words, *what* the system must accomplish. Other kinds of requirements (such as interface requirements, performance requirements, or reliability requirements) describe *how* the system accomplishes its functional requirements. Each functional requirement should be specified in a format similar to the following:

The section organizes functional requirements of the P-VPN Method Web Application according to their order of priority. The set requirements specify every function that makes the system functional.

1. Conduct Speed Test

The application provides a feature that enables users to start VPN speed tests. The application needs to evaluate download speed together with upload speed and ping time while users connect through a specific VPN protocol such as OpenVPN, WireGuard or Proton VPN.

2. View Speed Test History

Prior speed test results will automatically get saved by the application directly to the browser storage function. The application enables users to access their past test records through a well-formatted output.

3. Submit Contact Form

Users can submit messages through the application's Contact Us form that consolidates FormSubmit service with fields for first name, last name, email, subject, and message entry. Computer systems will transmit the form data through electronic mail to the recipient designated during submission.

4. View VPN Comparison

We have equipped the application with a Comparison area which displays extensive results about the examined VPN services. The system tracks speed performance alongside processor usage along with traffic processing capabilities and expenses.

5. Access "Contact Me" Button

The navigation bar contains a "Contact Me" button which launches the default email program of the user to initiate correspondence with the developer.

6. Some Cybersecurity Measures.

In a bid to ensure preserving of user interaction and system integrity, bareminum cybersecurity is enforced by the application. This has a CAPTCHA system inbuilt in the contact form to avoid spam and automated misuses. In addition, the speed test feature has created an option whereby a user can perform up to a maximum of five

tests within a ten-minute period. These controls are beneficial in lessening the server load, removal of abuse, and the dependability of the test results.

Short, imperative sentence stating highest ranked functional requirement.

2.1.1.1. Use Case Diagram

2.1.1.2. Requirement 1 <Name of requirement in a few words>

The heading of this section should read, e.g., “Requirement 1: User registration” or “Requirements 1: Participant takes test”

The system enables all users to start VPN speed tests. Application users can start VPN speed tests that will evaluate multiple essential network performance indicators.

Download speed

Upload speed

Ping time

The application will record network performance metrics during VPN connection through OpenVPN and WireGuard and Proton VPN protocols and additional user-selected protocols. Users can assess VPN performance through this capability under varied circumstances.

User tests their speed

2.1.1.3. Description & Priority

A description of the requirement and its priority. Describes how essential this requirement is to the overall system.

The required feature allows users to evaluate different VPN protocols by executing speed tests for download and upload operations as well as ping measurement. The tested results offer numerical evidence to enable users in selecting appropriate VPN services that match their specific requirements.

Priority: High

Core functional operation of the system depends on this feature. The application's main purpose is VPN comparison and accurate speed testing allows users to perform successful assessment.

2.1.1.4. Use Case

Each requirement should be uniquely identified with a sequence number or a meaningful tag of some kind.

Scope

The scope of this use case is to …….

Description

This use case describes the ……..

1

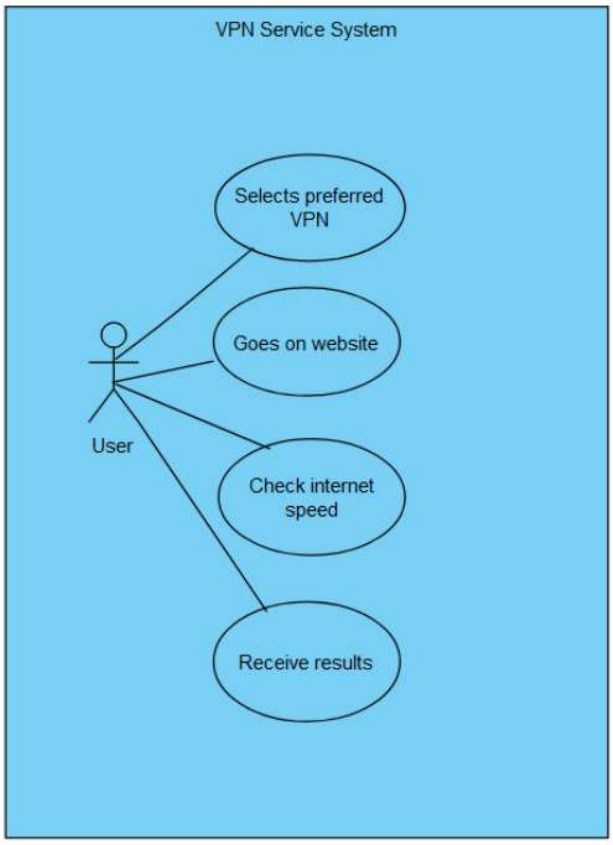
This use case describes when a user opens the webapp and uses the speed tester functionality on the web app.

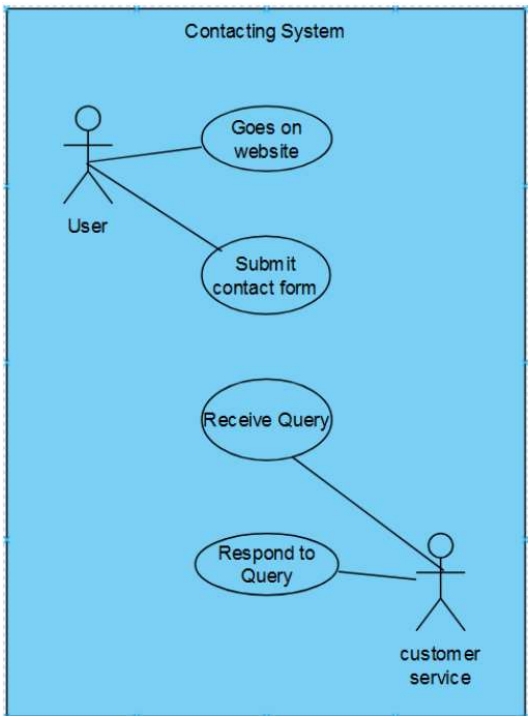
2

This use case describes when a user opens the webapp and uses the contact us functionality

Use Case Diagram

Diagram should highlight actors and uses cases……..





Flow Description

1

This use case will describe how a user will test their speed on the web app

2

This use case describes how a user will communicate with the developer.

Precondition

The system is in initialisation mode……..

1

The precondition for this use case is that the user must be connected to the internet.

2

The precondition for this use case is that the user must have contact details.

Activation

This use case starts when an <Actor>……

1

This use case starts when the user opens the webapp.

2

This use case starts when when the user opens the web app

Main flow

1. The system identifies the …….
2. The <Actor> ……...(See A1)
3. The system ……..(See E1)
4. The <Actor> …….

Alternate flow

A1 : <title of A1>

1. The system ……..
2. The <Actor> …….
3. The use case continues at position 3 of the main flow

1. The system identifies the User (U1) on the speed tester page, the user got to the speed tester page from most likely the home page.

2. The User (U1) presses "test your speed" button.
3. The system accepts the User (U1) request and gives them details to their connection speed
4. The user can see their results displayed

Exceptional flow

E1 : <title of E1>

4. The system ……..
5. The <Actor> …….
6. The use case continues at position 4 of the main flow

Termination

The system presents the next …….

The web app presents the user with an error message to prompts the user that their connection was "connection not established"

Post condition

The system goes into a wait state

If the user clicks the "test your speed" and it doesn't work, the screen will display a wait state until the webpage can get a response.

List further functional requirements here, using the same structure as for Requirement1.

2.1.2. Data Requirements

2.1.3. User Requirements

2.1.4. Environmental Requirements

2.1.5. Usability Requirements

2.2. Design & Architecture

Describe the design, system architecture and components used. Describe the main algorithms used in the project. (Note use standard mathematical notations if applicable).

The VPN Comparison Web App operates as a client-side browser application made with HTML CSS and JavaScript. The application implements a basic MVC structure based on the three components.

The application saves its speed test information inside the localStorage model.

View: HTML/CSS-based interface.

The application logic together with user interactions are managed by JavaScript.

The application has no server backend system since all processing takes place through local dynamics or third-party service utilization.

Components

This module performs tests for download/upload speed and ping through native JavaScript timing mechanisms.

The History Viewer application both stores and presents speed test results from the past.

Rate Limiting Mechanism: The number of speed tests that the user can take is limited to 5 tests during a 10-minute period through a cybersecurity protocol incorporated into the speed test function. This will prevent misuse and browser resource exhaustion.

Comparison Page: Displays VPN performance comparisons.

The Contact Form depends on FormSubmit for sending emails through the system.

CAPTCHA Validation: A CAPTCHA is used within the contact form to prevent spam from being inputted and abuse from being carried out as only human will be allowed submission.

The IP leakage test appears as a Wireshark screenshot with blurred content in order to show the results of the Kill Switch Test.

An architecture diagram may be useful. In case of a distributed system, it may be useful to describe functions and/or data structures in each component separately.

2.3. Implementation

Describe the main algorithms/classes/functions used in the code. Consider to show and explain interesting code snippets where appropriate.

The project is written primarily in **JavaScript HTML and CSS**, with logic organized into functional modules. Below are the key functions and their purposes: [1] [2]

1. Speed Test Logic

How does speed test work?

1. Latency Test (Ping)

- **Function:** measureLatency()
- **How it works:**

  It sends a small request to Cloudflare's 1.1.1.1 service.

  It records the time before and after the request.

  The difference in time is the latency (ping) in milliseconds (ms).

- **Why use 1.1.1.1?**

  Cloudflare’s DNS is very fast and reliable for measuring response time

2. Download Speed Test

- **Function:** measureDownloadSpeed()
- **How it works:**

  It attempts to download test files from Cloudflare, Hetzner, and OVH which are each 10mb

  Measures the time taken to download a file.

  Calculates speed in megabits per second (Mbps) using:

Speed=File Size in BitsDownload Time in Seconds\text{Speed} = \frac{\text{File Size in Bits}}{\text{Download Time in Seconds}}Speed=Download Time in SecondsFile Size in Bits

- **Failsafe:** If one file fails, it tries another.
- **Why multiple servers?**

  To get the most accurate test in case a server is slow or down.

3. Upload Speed Test

- **Function:** measureUploadSpeed()
- **How it works:**

  Creates a dummy file (2MB) filled with 'A' characters.

  Sends it to Cloudflare's upload test server (https://speed.cloudflare.com/__up).

  Measures the time taken to upload.

  Uses the same speed formula as the download test.

4. Speed Calculation

- **Function:** calculateSpeed(size, start, end)
- **How it works:**

  Converts bytes → bits (1 byte = 8 bits).

  Divides by the time taken.

Returns speeds in bps (bits per second), kbps (kilobits per second), and Mbps (megabits per second).

5. Event Listener & UI Updates

- **When the user clicks "Start Speed Test":**
  Runs all three tests.

  Displays results inside the elements in the HTML.

The speed test calculates download/upload speed and ping time using the browser's performance timers.

```
/ set placeholder text while test is running
ownloadOutput.innerHTML = "Testing...";
ploadOutput.innerHTML = "Testing...";
atencyOutput.innerHTML = "Testing...";

ry {
   // get the latency and display it
   let latency = await measureLatency();
   latencyOutput.innerHTML = latency !== "N/A" ? `${latency} ms` : "Error";

   // get download speed and display
   let downloadSpeed = await measureDownloadSpeed();
   downloadOutput.innerHTML = downloadSpeed.mbps > 0 ? `${downloadSpeed.mbps} Mbps` : "Error";

   // get upload speed and display
   let uploadSpeed = await measureUploadSpeed();
   uploadOutput.innerHTML = uploadSpeed.mbps > 0 ? `${uploadSpeed.mbps} Mbps` : "Error";

   // if all goes well, save results to localStorage
   if (latency !== "N/A" && downloadSpeed.mbps > 0 && uploadSpeed.mbps > 0) {
       saveTestResult(downloadSpeed.mbps, uploadSpeed.mbps, latency);
```

```
// Function to measure latency
async function measureLatency() {
    try {
        let startTime = performance.now();
        // using 1.1.1.1 to ping for latency
        await fetch("https://1.1.1.1/cdn-cgi/trace", { cache: "no-store" });
        let endTime = performance.now();
        return (endTime - startTime).toFixed(2); // return latency
    } catch (error) {
        console.error("Latency test failed:", error);
        return "N/A"; // fallback if something goes wrong
    }
}

// Function to measure download speed
async function measureDownloadSpeed() {
    // list of test files from different servers
    let testFiles = [
        "https://speed.cloudflare.com/__down?bytes=10485760",
        "https://speed.hetzner.de/10MB.bin",
        "https://proof.ovh.net/files/10Mb.dat"
```

```
// Function to calculate speed based on size and duration
function calculateSpeed(size, start, end) {
    let duration = (end - start) / 1000;
    let bits = size * 8;

    return {
        bps: (bits / duration).toFixed(2),
        kbps: (bits / (1024 * duration)).toFixed(2),
        mbps: (bits / (1024 * 1024 * duration)).toFixed(2)
    };
}
```

2. Ping Test Function

A simple ping is calculated using round-trip time of an HTTP request:

```
// Function to measure latency
async function measureLatency() {
    try {
        let startTime = performance.now();
        // using 1.1.1.1 to ping for latency
        await fetch("https://1.1.1.1/cdn-cgi/trace", { cache: "no-store" });
        let endTime = performance.now();
        return (endTime - startTime).toFixed(2); // return latency
    } catch (error) {
        console.error("Latency test failed:", error);
        return "N/A"; // fallback if something goes wrong
    }
}
```

Makes sure user can only run 5 speed tests every 10 mins:

First, it receives the current time in milliseconds.

Then it pulls all of the previously saved speed test times from these local storage. If there aren't any, it uses empty list.

It screens these by removing all of those that are more than 10 minutes old.

If there are at least five or more tests in that time window it shows an alert message and stops the user running another test.

```
// Function to check if the user can run a speed test (limit: 5 tests per 10 minutes)
function canRunSpeedTest() {
    const now = Date.now(); // current time in milliseconds
    const saved = localStorage.getItem("speedTestTimestamps");
    const timestamps = saved ? JSON.parse(saved) : [];

    // Keep only timestamps from the last 10 minutes (600,000 ms)
    const recent = timestamps.filter(ts => now - ts < 10 * 60 * 1000);

    // If the user has already done 5 tests, block the test
    if (recent.length >= 5) {
        alert("You've reached the limit of 5 speed tests in 10 minutes. Please wait a while
        return false;
    }
```

Save & Load History (Local Storage)

Results are stored in localStorage for later viewing:

```
// Function to save results to localStorage
function saveTestResult(downloadMbps, uploadMbps, latencyMs) {
    // get existing history or start a new array
    const history = JSON.parse(localStorage.getItem("speedTestHistory")) || [];

    // add new test result to history
    history.push({
        date: new Date().toLocaleString(),
        download: `${downloadMbps} Mbps`,
        upload: `${uploadMbps} Mbps`,
        latency: `${latencyMs} ms`
    });

    // save updated history back to localStorage
    localStorage.setItem("speedTestHistory", JSON.stringify(history));
}
```

ReCAPTCHA:

Users are asked to complete a reCAPTCHA before submitting their form for security purposes

```
// Validation, sanitization, and reCAPTCHA on submit
document.querySelector("form").addEventListener("submit", function (e) {
  const fNameInput = document.querySelector(".first-name");
  const lNameInput = document.querySelector(".last-name");
  const emailInput = document.querySelector(".email");
  const subjectInput = document.querySelector(".subject");
  const messageInput = document.querySelector(".message");

  const fName = sanitizeInput(fNameInput.value.trim());
  const lName = sanitizeInput(lNameInput.value.trim());
  const email = sanitizeInput(emailInput.value.trim());
  const subject = sanitizeInput(subjectInput.value.trim());
  const message = sanitizeInput(messageInput.value.trim());

  const emailPattern = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;
  const recaptchaResponse = grecaptcha.getResponse(); // reCAPTCHA check

  if (!fName || !lName || !email || !subject || !message) {
    alert("Please fill in all fields.");
    e.preventDefault();
```

Makes sure user completes captcha before submitting:

```
//If user tries to submit without completing reCAPTCHA
    if (!recaptchaResponse) {
      alert("Please complete the reCAPTCHA.");
      e.preventDefault();
      return;
    }
```

Contact Form Submission

FormSubmit is used to send form data via email:

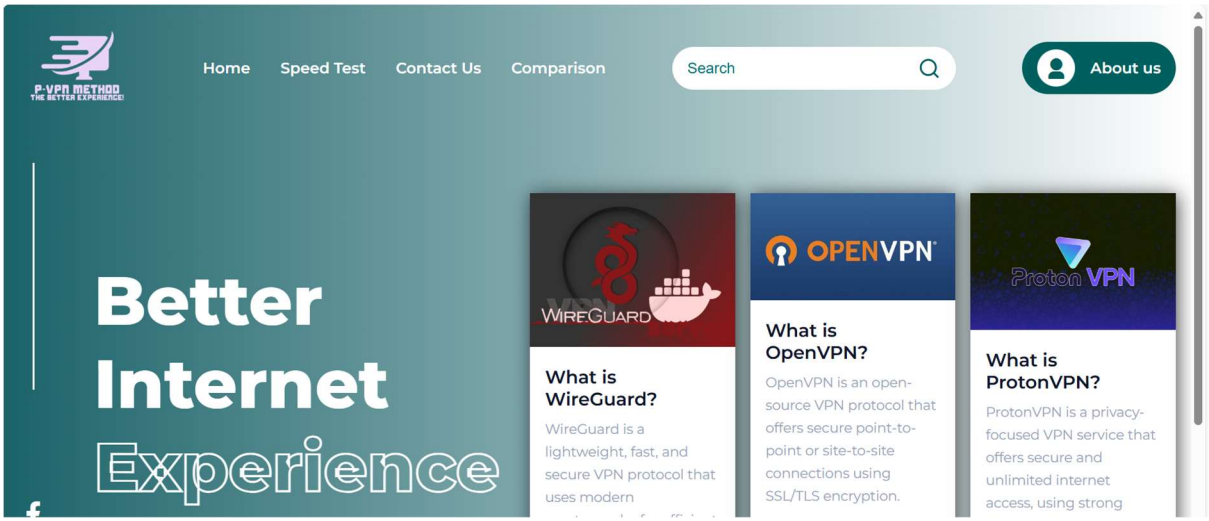
```
<main>
    <!-- Form submit so users can contact us-->
    <form action="https://formsubmit.co/pdawg9073@gmail.com" method="POST" class="contact-container">
      <input type="hidden" name="_next" value="http://localhost:8000/contactus.html">
```

2.4. Graphical User Interface (GUI)

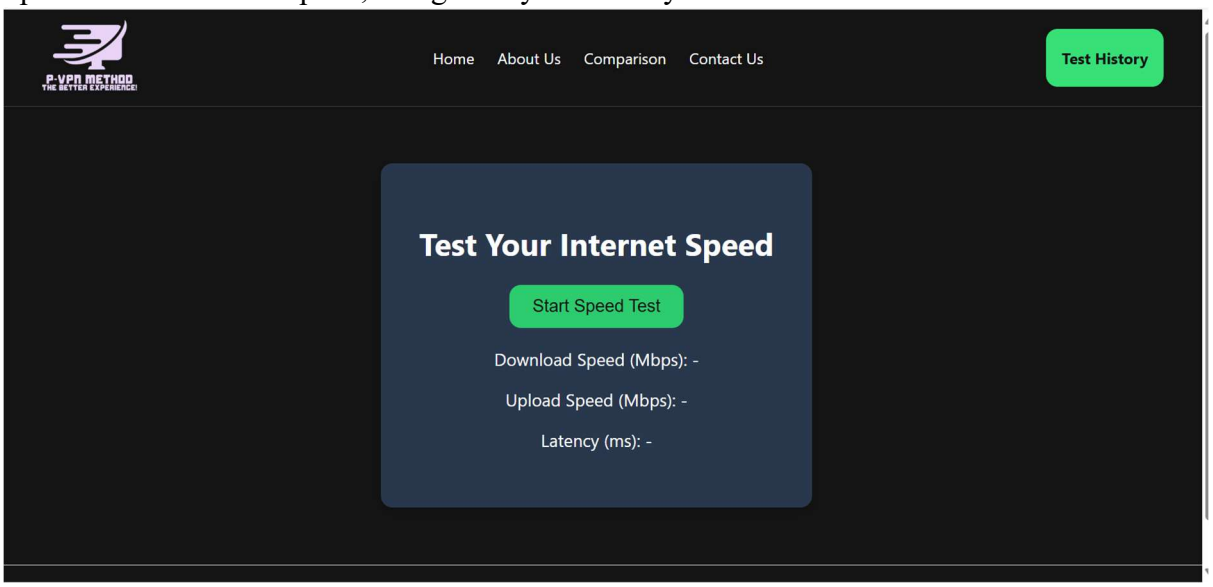
Provide screenshots of key screens and explain what can be seen in each one.

[3] [4] [5] [6]

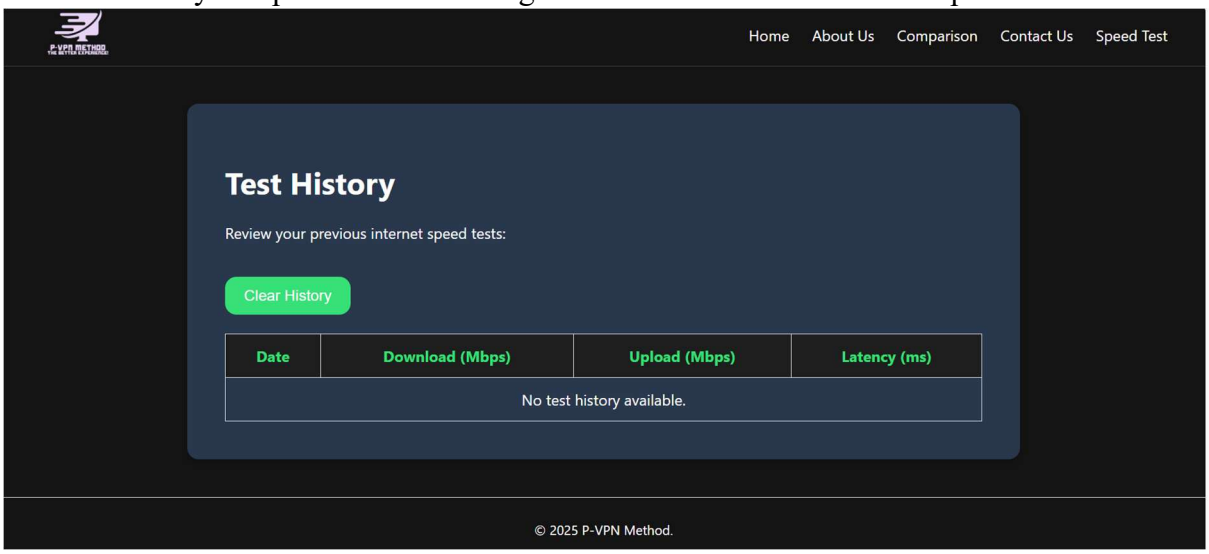
Here is my homepage where users can see more information on the VPNs, where then can then also access other pages using the navbar links.



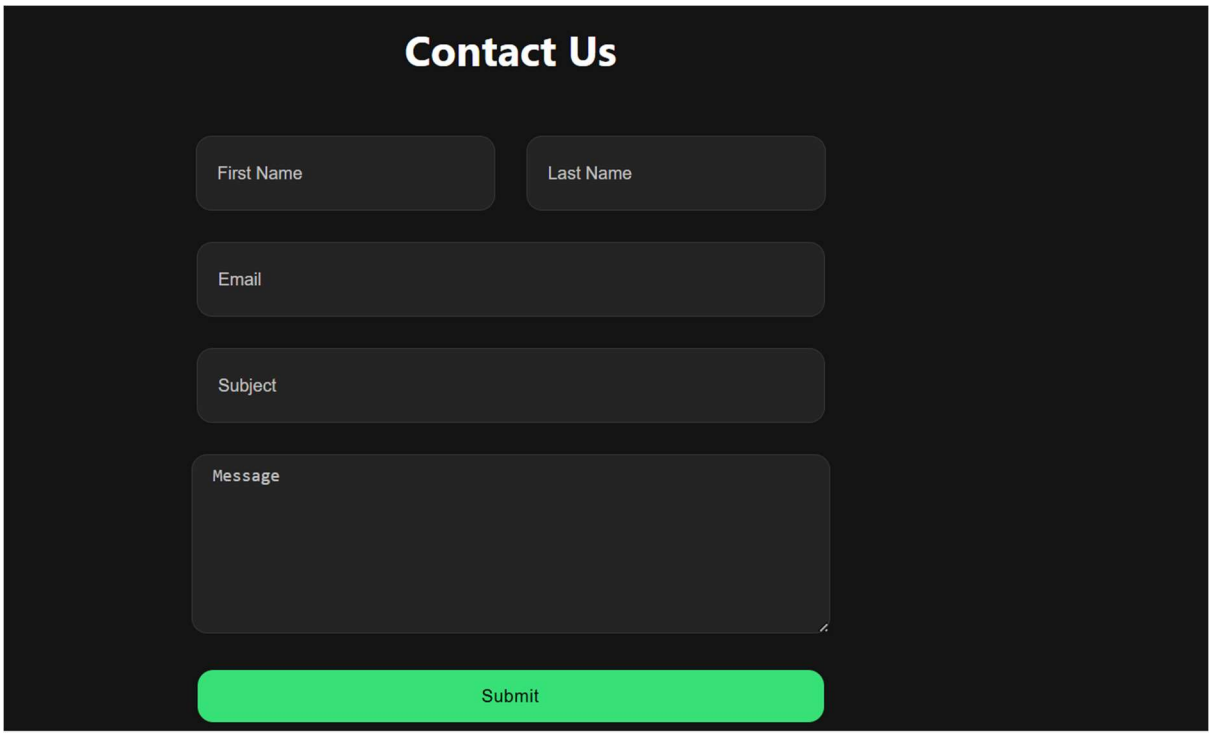
Here is my speed test page where once user clicks start speed test, it will calculate your upload and download speed, along with your latency.



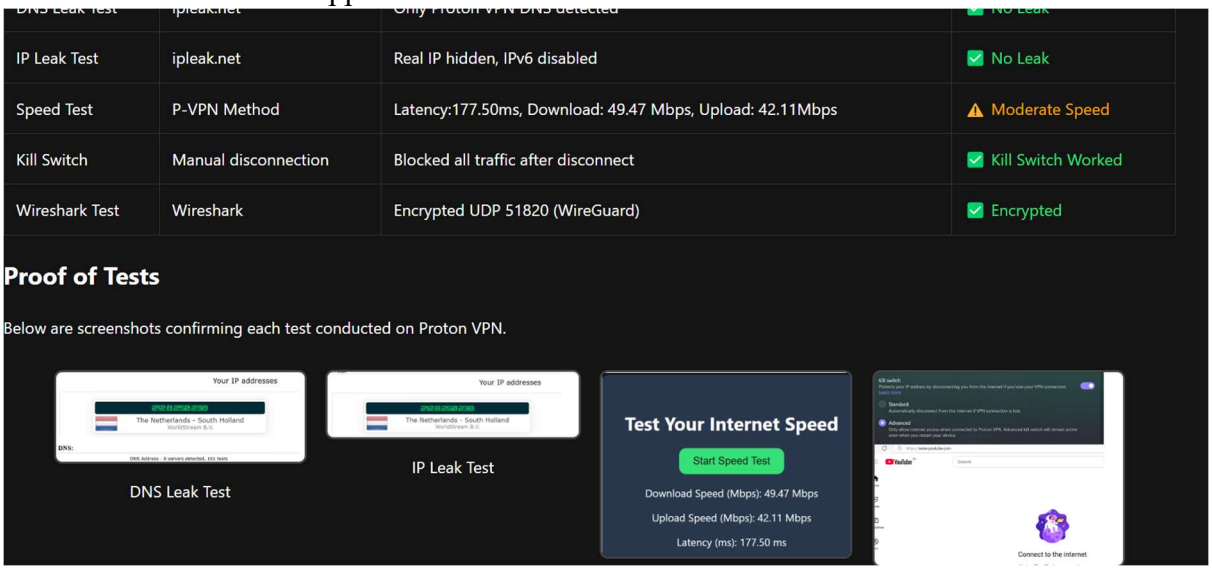
Here is my test history page where users can access form the speed test page to see their recent tests they had performed including the time date and minute it was performed.



Here is where my contact us page incase a suser wasnts to send me a message whilst still being annonymous by using form submit.



Here we have out comparisons page where user can view the 5 different tests that were carried out to make this application.



2.5. Testing

[7]

Describe any testing tools, test plans and test specifications used in the project. Provide evidence for and results of all Unit, Integration and End User testing that is carried out.

Testing Tools, Plans, and Specifications

Framework tests of the VPN Comparison Web Application demonstrated proper functionality alongside user-friendly reliable output of the site. My unit testing and user

interface testing used manual third-party service integration as a sequence of integration testing. Browser tools supported this evaluation process.

Testing Tools Used

The analysis of HTML/CSS layout and JavaScript debugging and screen size testing relied on Chrome DevTools. Ipleak.net provided essential tools to evaluate VPN security elements by detecting IP and DNS leakages. Through Wireshark I examined packet traffic to verify VPN encryption through monitoring for protected packets sent on VPN protocol ports. The FormSubmit testing tool provided contact form simulation with email sending capabilities through testing that did not need server-based code functions. I conducted tests of localStorage to verify the correct storage of user results in speed tests.

Test Plan and Specification Overview

I conducted unit examinations for all JavaScript components. I manually verified that:

The speed test logic precisely determined the speed values for downloads and uploads.

The application correctly accessed and retrieved test results through localStorage storage.

The modal image requires correct content and caption at initialization.

The contact form checked the validity of user entries while properly transferring users to the next page following submission.

The application's different sections were integrated for testing purposes using the Integration testing method. For example:

Users could view their speed test results in the user interface while the system also recorded them for later reference in their past tests.

When users clicked on proof images the modal image viewer functioned correctly to display them.

The website navigation bar operated smoothly among all pages with no missing links.

My simulations of end-user tests involved the use of Chrome, Firefox and Safari browsers and laptops. I performed operations on the site like a regular visitor would do by moving between pages and running speed tests and reviewing VPN comparisons before sending the contact form. The testing procedure enabled me to spot existing layout and usability problems.

Results of Testing

Unit tests satisfied all their expected conditions. Both speed testing and modal image functions in JavaScript operated correctly along with proper storage and retrieval of test results through localStorage. FormSubmit received proper input data which let the system redirect users back to the form page after the form completion.

The integration tests confirmed smooth operation of every component combination between the speed test functionality and the modal image gallery with site navigation buttons. Any screenshot from the proof gallery opened the desired high-resolution picture with its associated caption and page change operations caused no technical problems.

The end-user testing process demonstrated that I successfully finished each task while operating from desktops. The application maintained interactive responses with legible content presentation through every screen size which was reflected across all browsers and operating systems. The system evaluated through testing showed both technical success and user-friendly functionality based on gathered performance data.

Speed test after clicking start speed test:

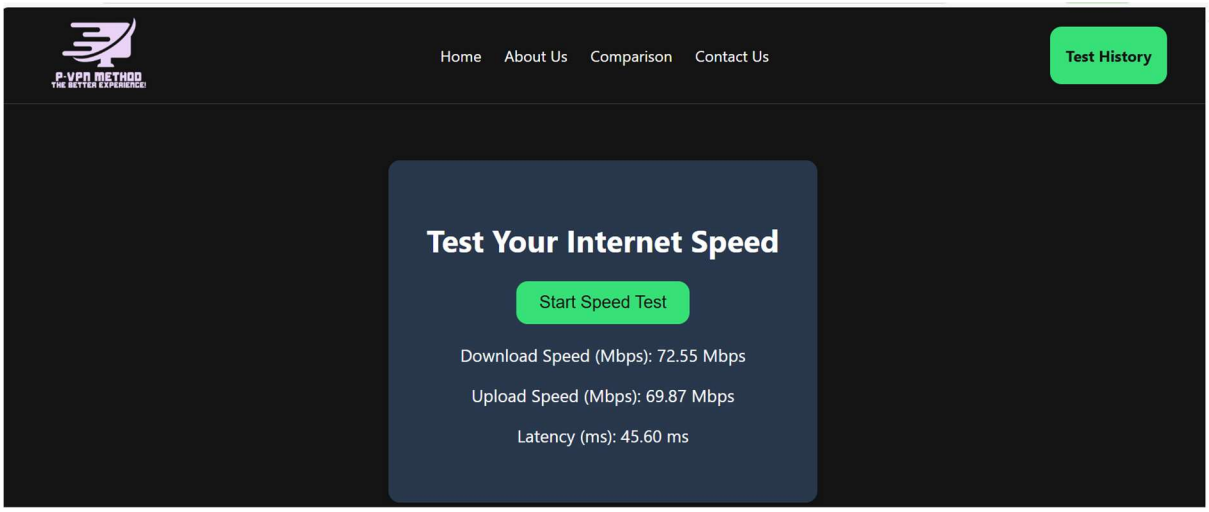
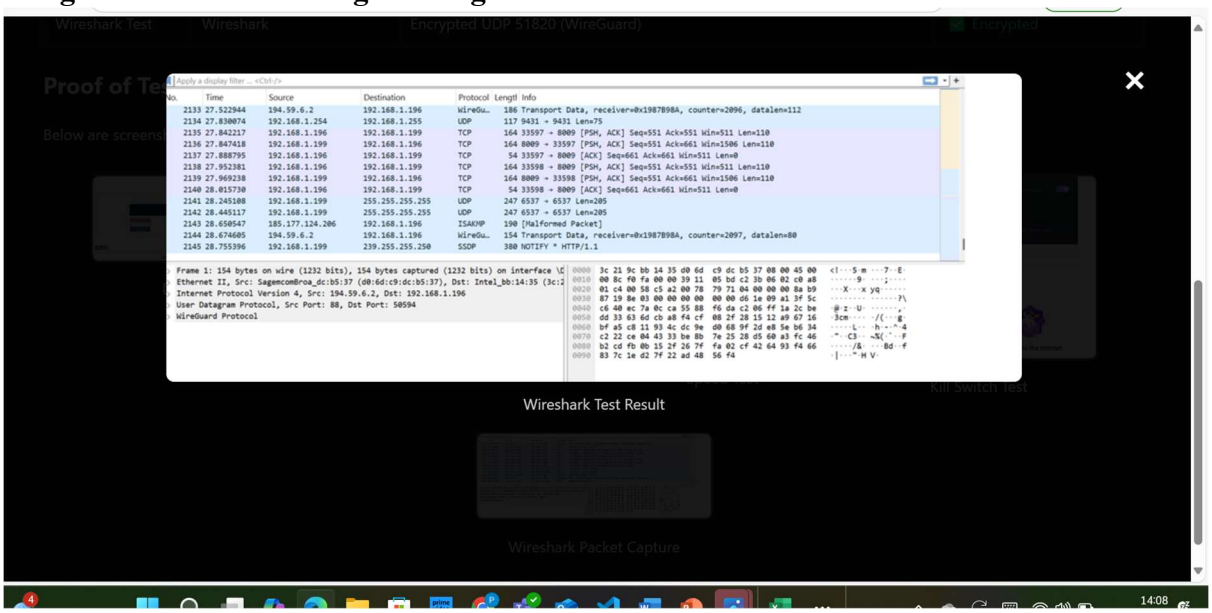
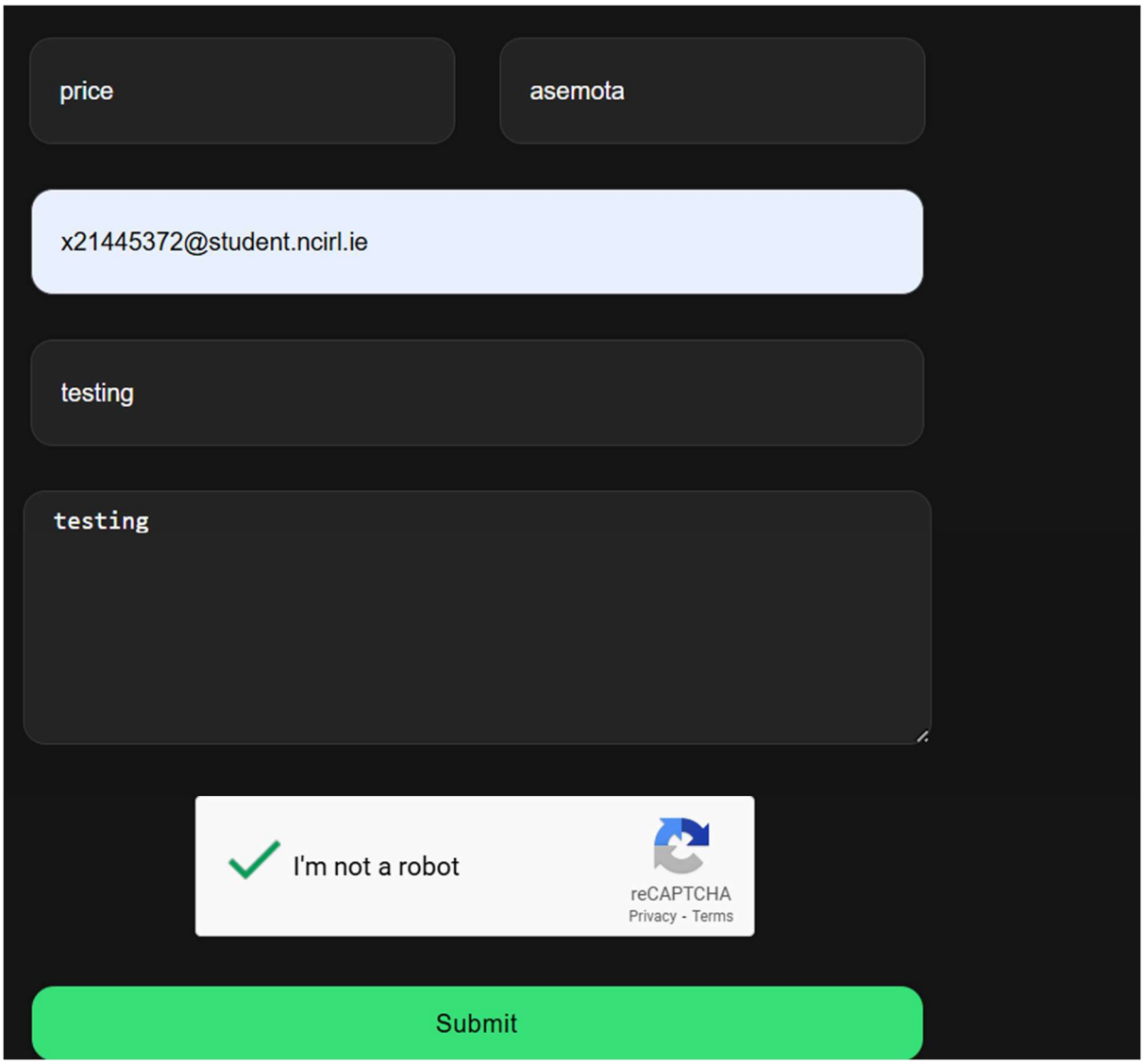


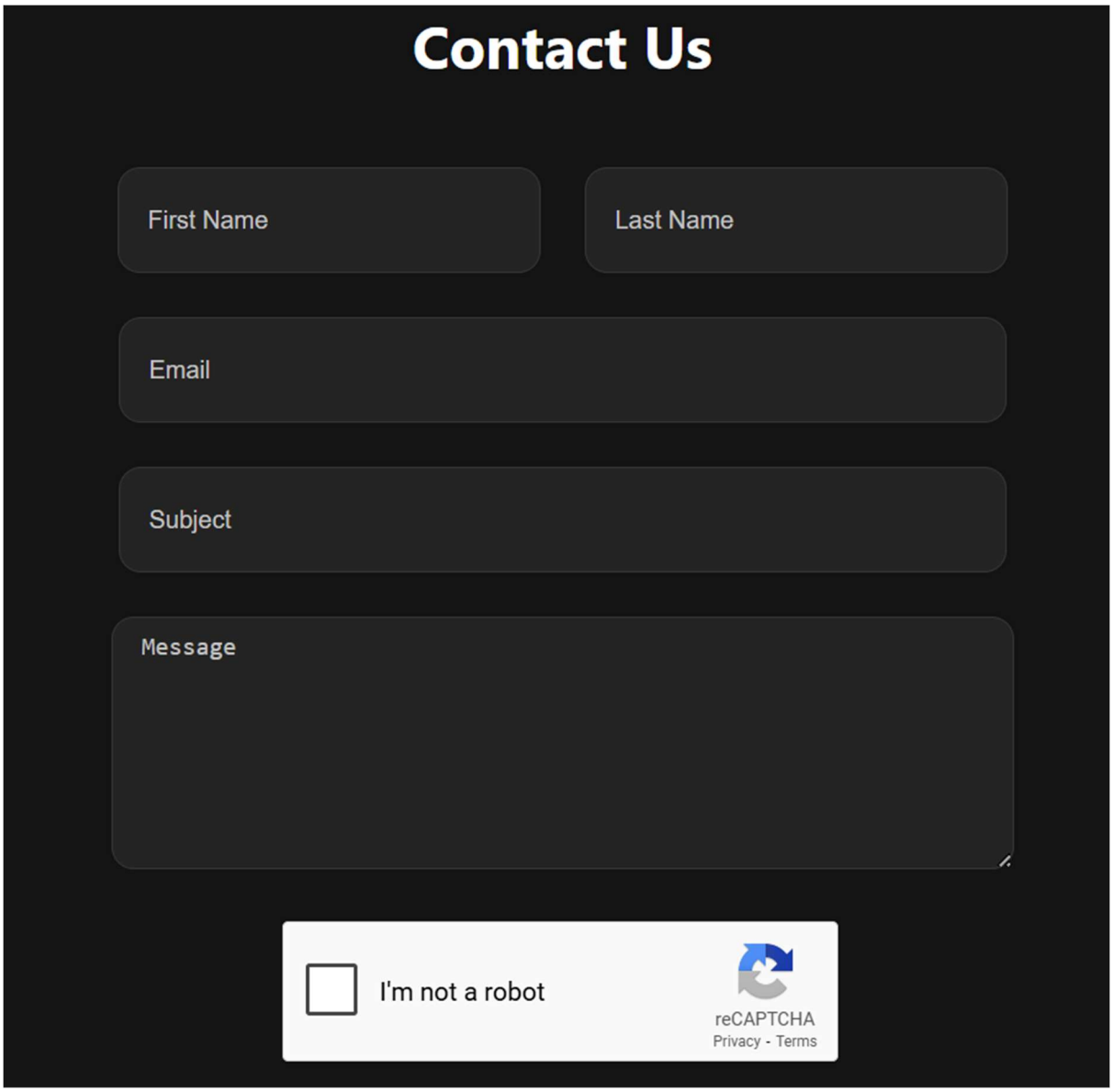
Image of test after clicking on image:



Contact us form before submitting:



After submission:



These original screenshots are included for grading purposes only. They are not shown in the public-facing web app to avoid leaking sensitive IP information.

[7]

1. DNS Leak Test

- **Purpose: Check if your DNS requests are leaking to your ISP instead of going through the VPN.**
- **ipleak.net**

---

2. IP Leak Test (IPv4, IPv6,)

- **Purpose: Verify that your real IP address is not exposed.**
- **Tool: ipleak.net**

---

3. Speed Test

- **Purpose: Measure VPN performance in terms of:**
  Ping (latency)

  Download speed

  Upload speed
- **My Speed test on my webapp**

---

4. Kill Switch Test

- **Purpose: Test if your VPN blocks internet traffic when the VPN connection drops.**
- **Tool: whatismyipaddress.com**
- **Method:**
  1. **Enable kill switch in settings.**
  2. **Disconnect VPN manually or disable network adapter temporarily.**
  3. **Try browsing — should fail if working properly.**

---

5. Wireshark Packet Capture

- **Purpose: Verify encrypted traffic by capturing packets on your interface and checking for:**
  No readable data

  VPN protocol in use (e.g., UDP 1194 for OpenVPN, UDP 51820 for WireGuard)

DNS LEAK

Why: Ensures that Proton VPN is securing your DNS queries and not leaking them to your ISP.

How I'm testing:

ipleak.net.

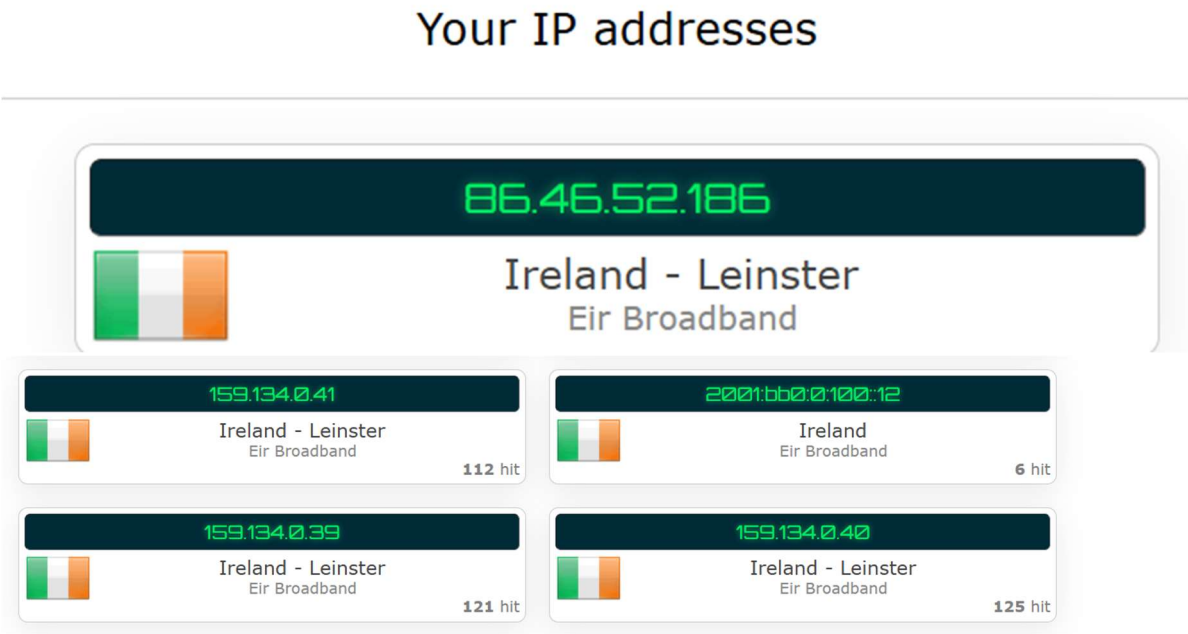
Connect to VPN, then run the test.

If you see your real ISP listed, that's a DNS leak.

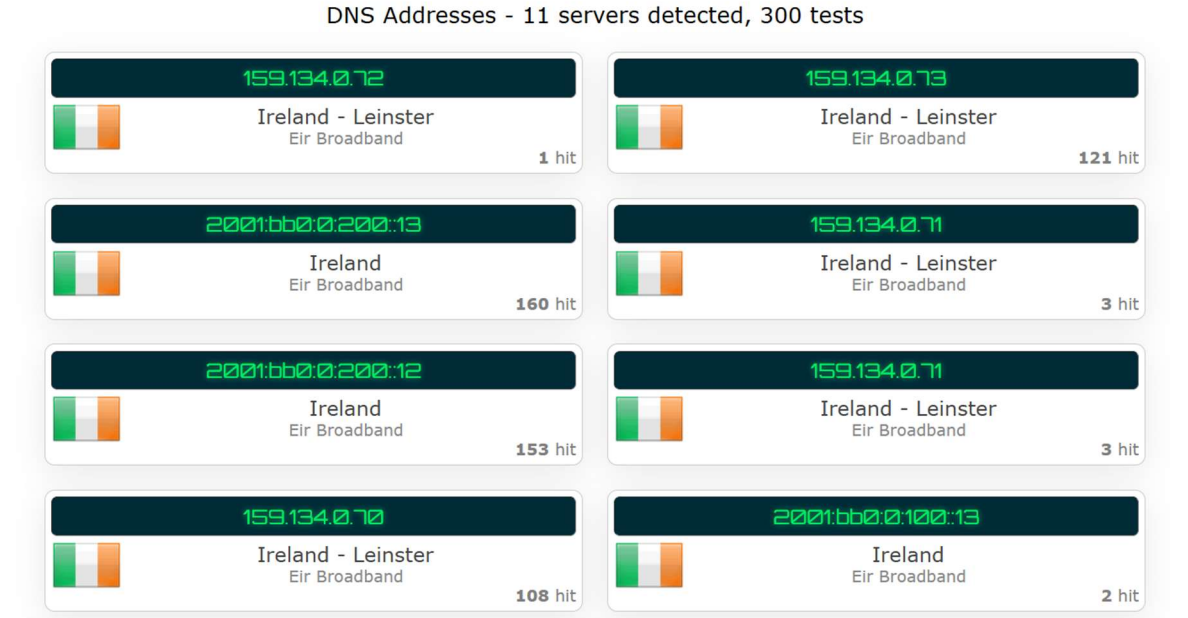
ProtonVPN:

Without VPN:

My IP adress

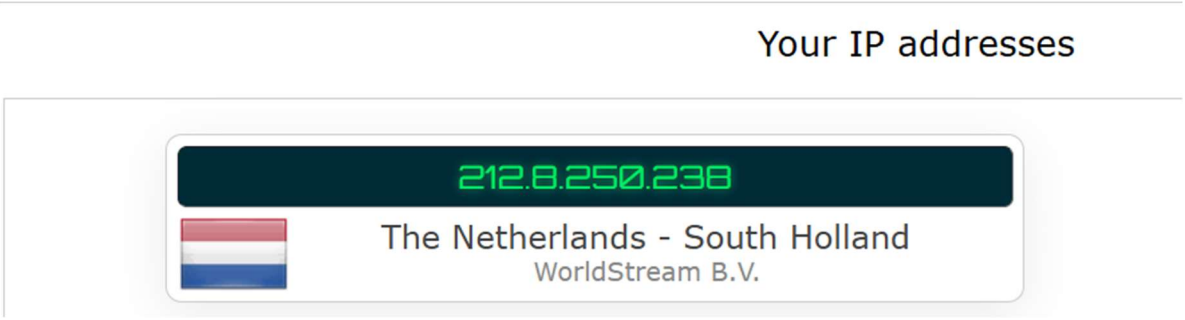


My DNS address, location and host names:

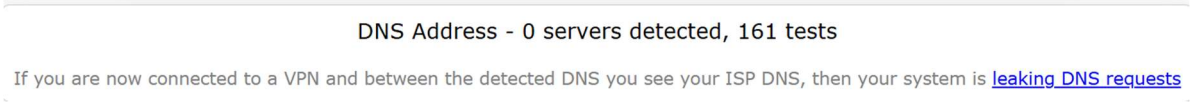


With VPN:

IP:



DNS:



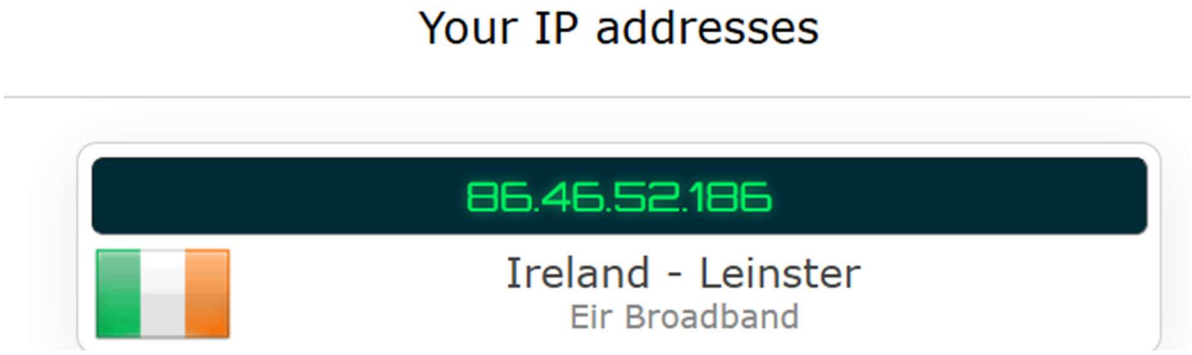
Result:

Works!

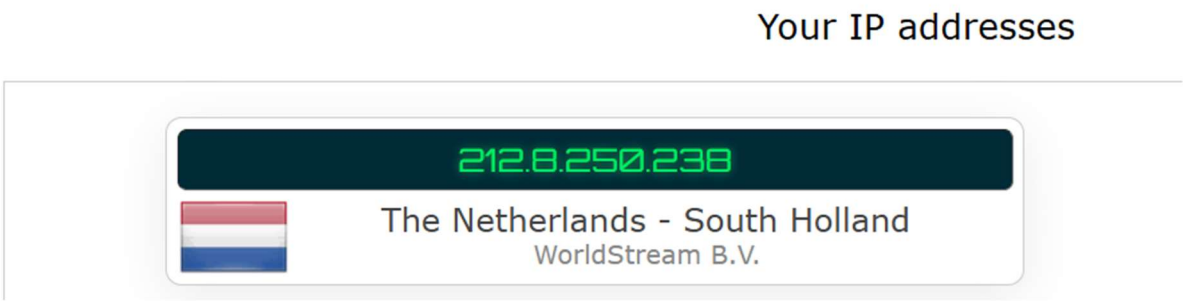
Proton VPN is routing all DNS traffic through their own secure servers, not my ISP's DNS servers.

IP TEST LEAK

Without



With:



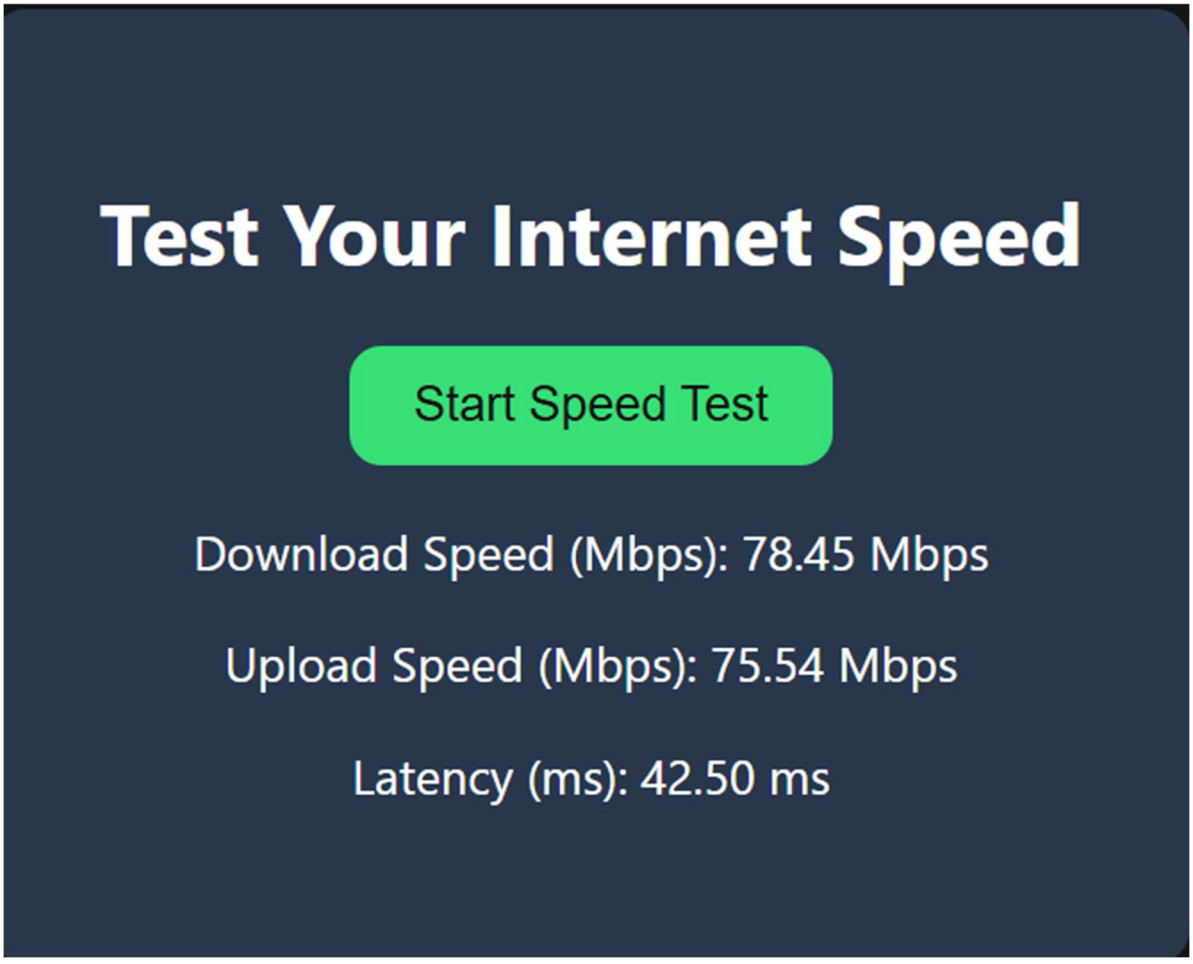
RESULTS:

Works!

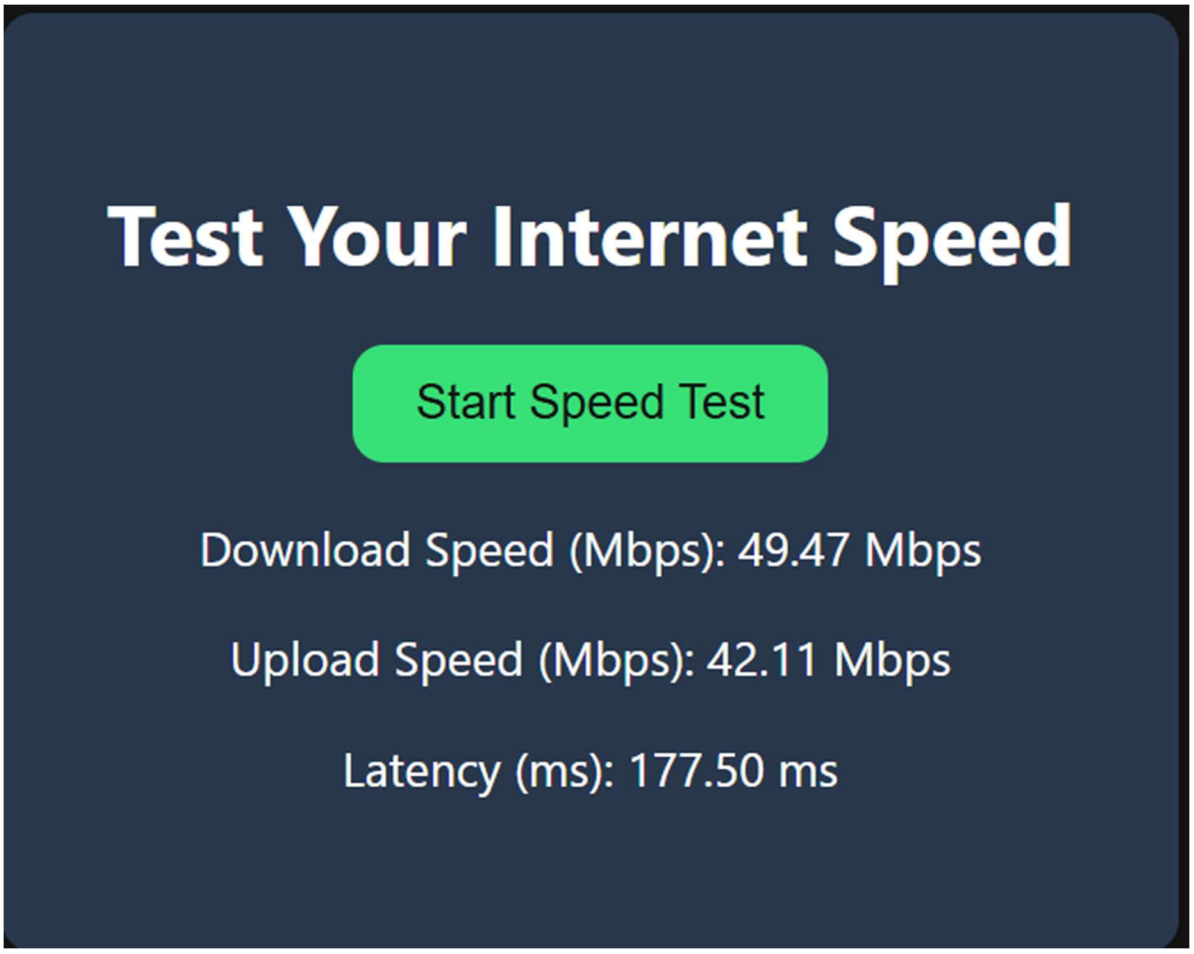
It shows a different IP address meaning it works

SPEED TEST

Without VPN:



With VPN:

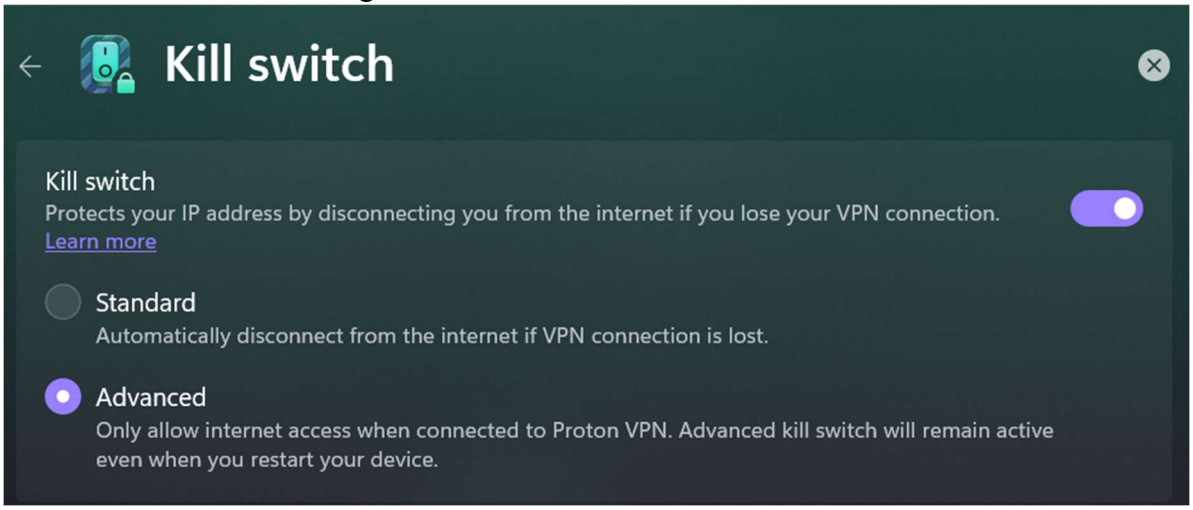


Upload Speed dropped slightly: That's normal

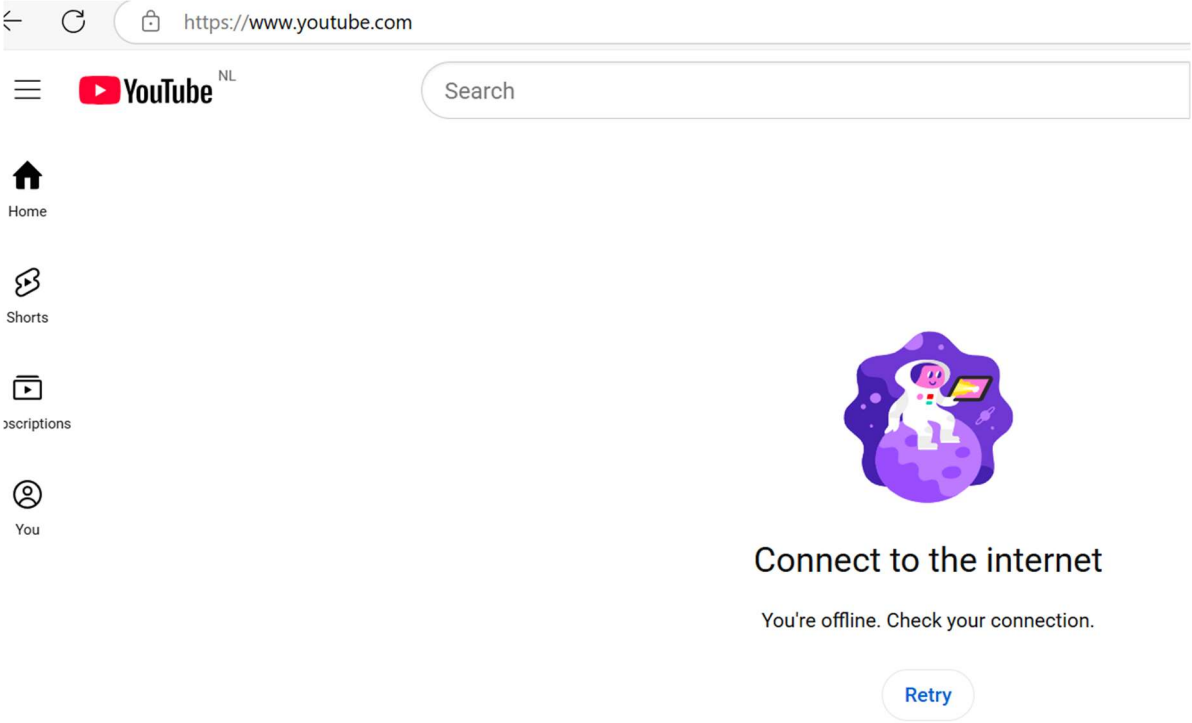
Latency increased: Expected, since my traffic is now being routed through Proton VPN servers.

KILL SWITCH

Turn on kill switch in settings.



After ending the proton VPN task in task manager, I wasn't able to reconnect to any website on my browser meaning the kill switch was a pass.



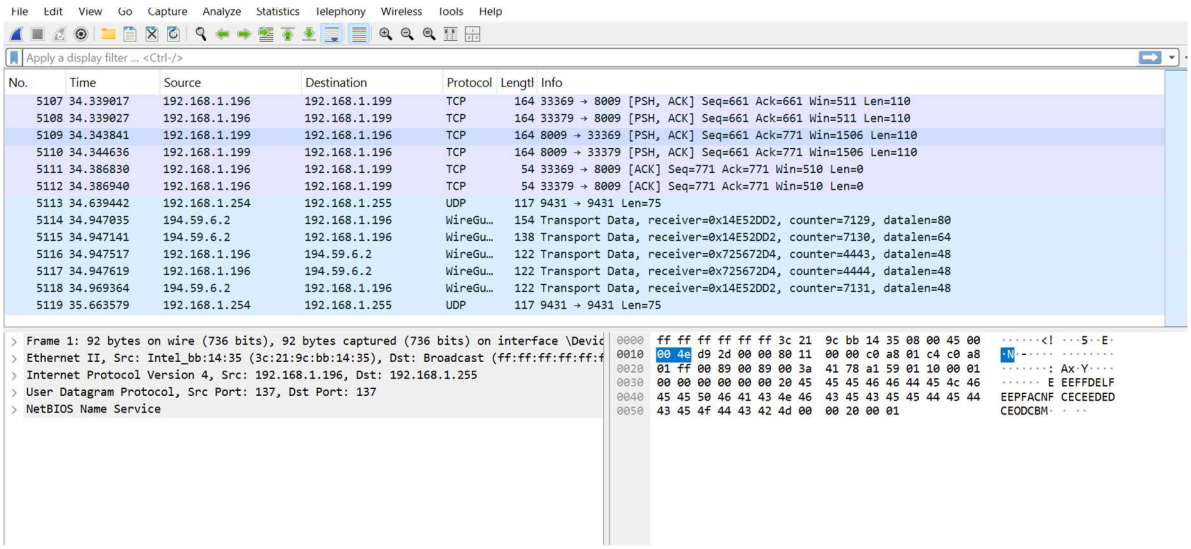
Status: Pass

Description:
After force-closing the Proton VPN app with Advanced Kill Switch enabled, internet access wasn't still active, indicating that the kill switch did function as intended.

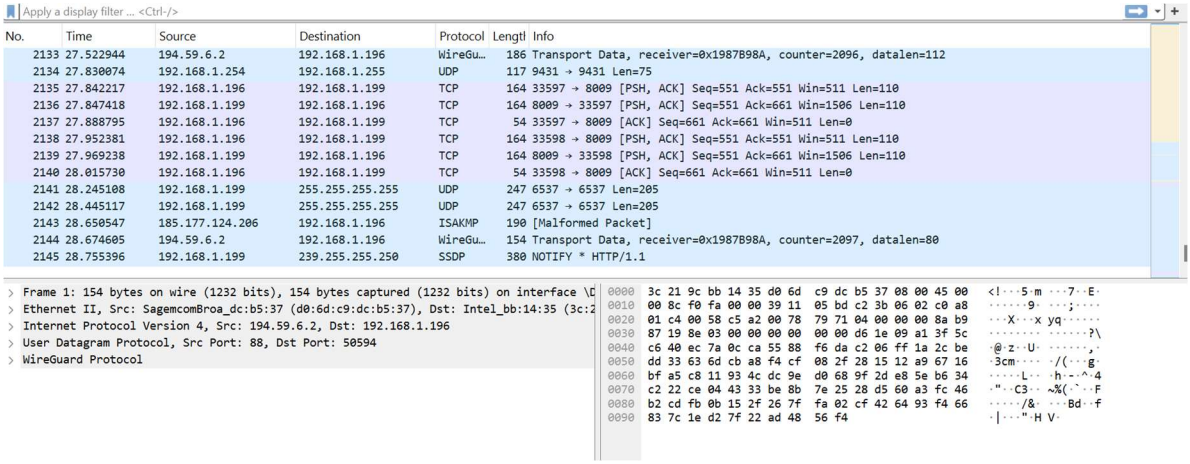
Wireshark Packet Capture:

Not connected to VPN:

Element	Explanation
192.168.1.x IP addresses	These are your **local/private IPs**, which are expected when not using a VPN.
Protocols like TCP, UDP, NetBIOS, and WireGuard	This is a mix of regular local traffic and traffic trying to initiate WireGuard — which is OK even though I'm not connected to it
No VPN protection active	This screenshot shows the device's real local network activity, which below will be contrasted with VPN-protected traffic.



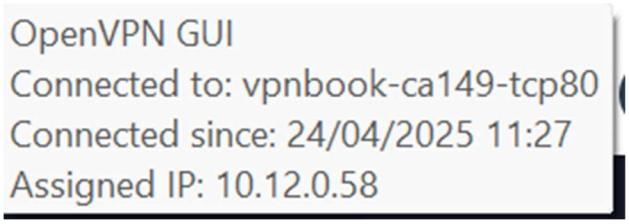
Connected to VPN:



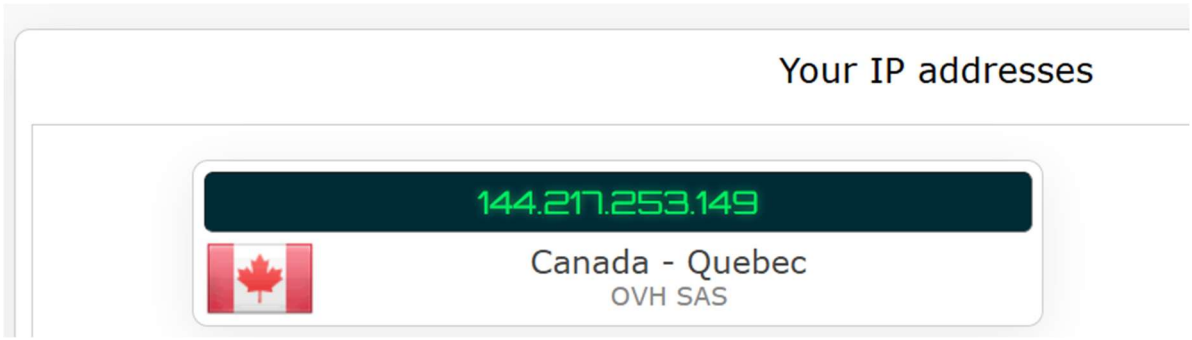
Result!

In the first Wireshark capture, where no VPN was used, traffic reveals my real local IP address (192.168.1.199) along with NetBIOS and SSDP packets, making my device easily fingerprint able and vulnerable on a public network. In contrast, the second capture shows Proton VPN in use with WireGuard protocol. The only visible IPs are those of the VPN servers, and the packet contents are fully encrypted, proving that my identity, data, and DNS queries are being tunnelled securely and anonymized from potential attackers on the same network.

OpenVPN:



IP TEST LEAK



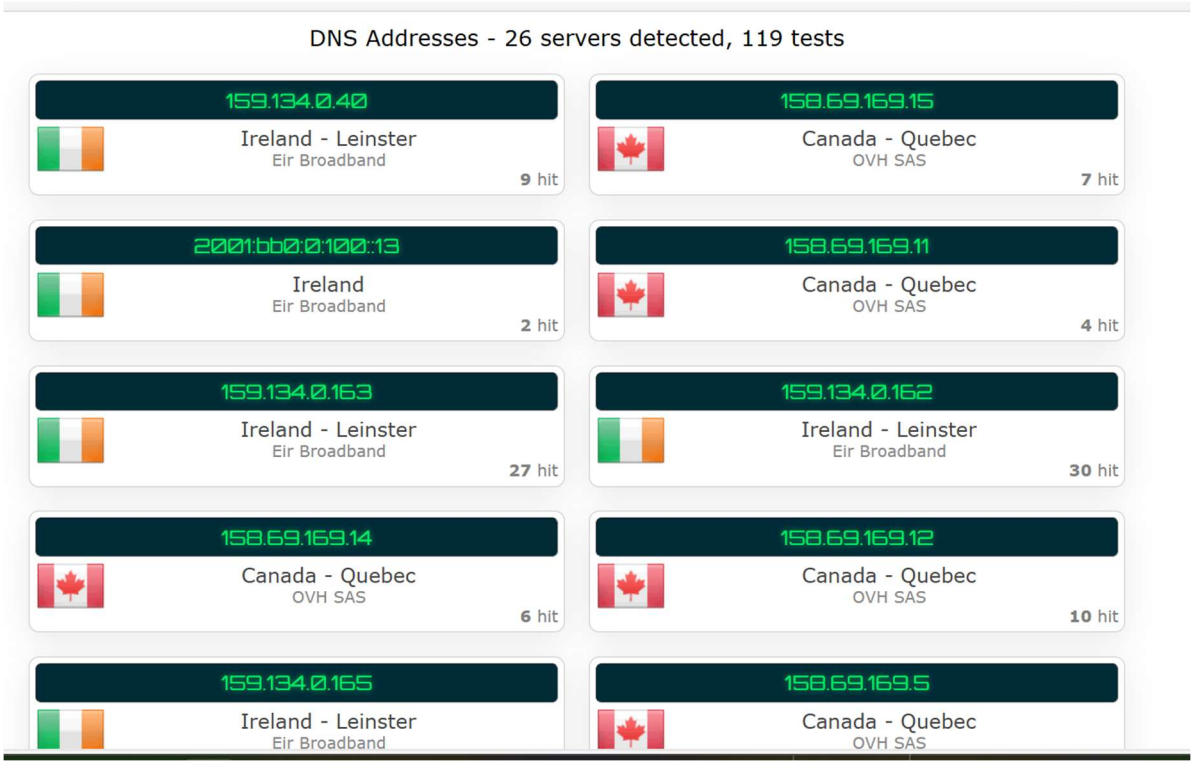
Result!

Works!

It shows a different IP address meaning it works**.**

DNS LEAK

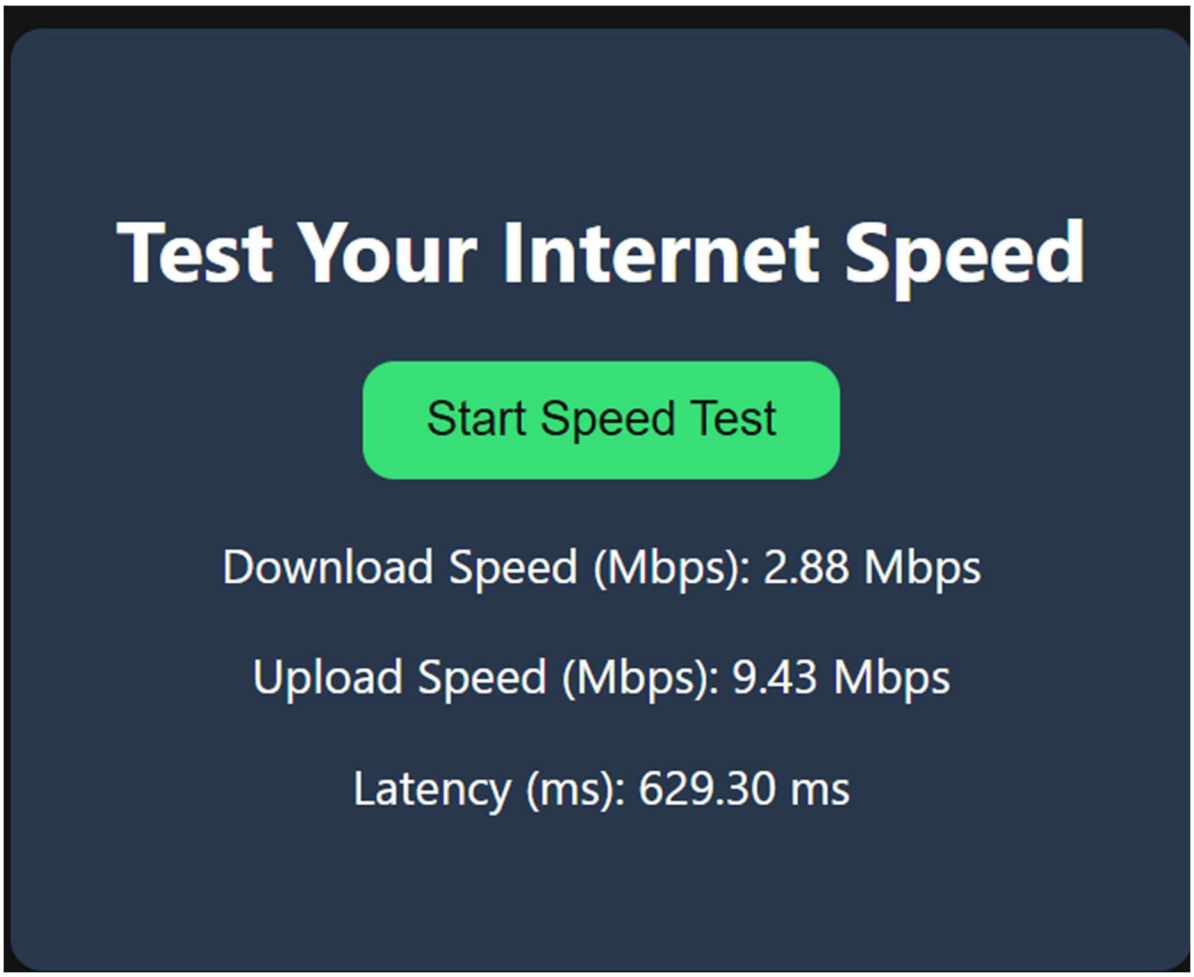
My DNS address, location and host names:



Result!

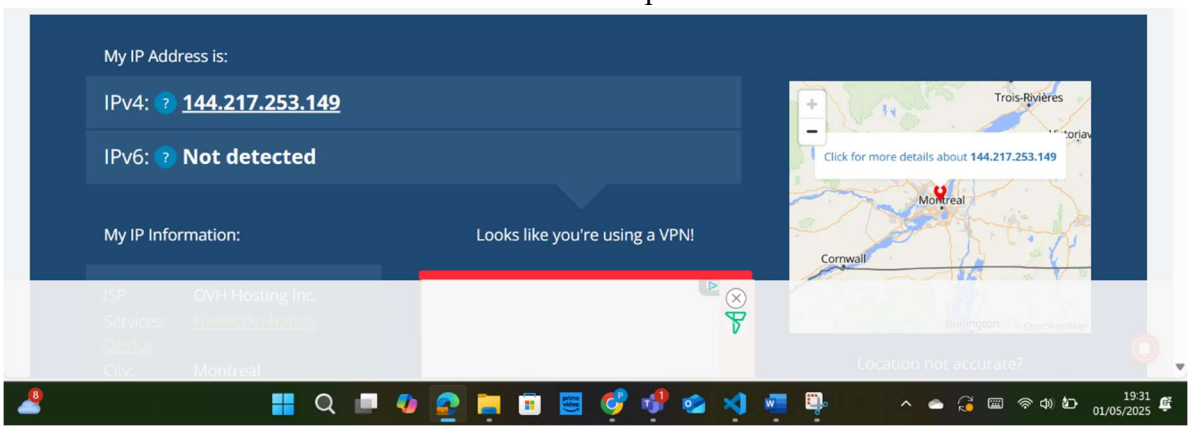
Open VPN is routing all DNS traffic through their own secure servers, not my ISP's DNS servers.

SPEED TEST

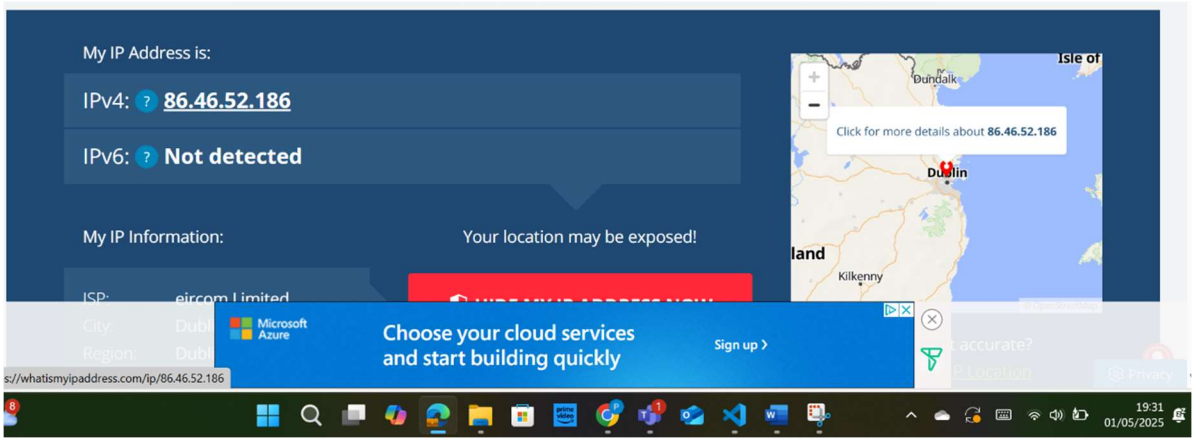


Kill Switch:

Here is what the browser looks like before I attempt the kill switch test



When I disconnect and refresh very quickly:

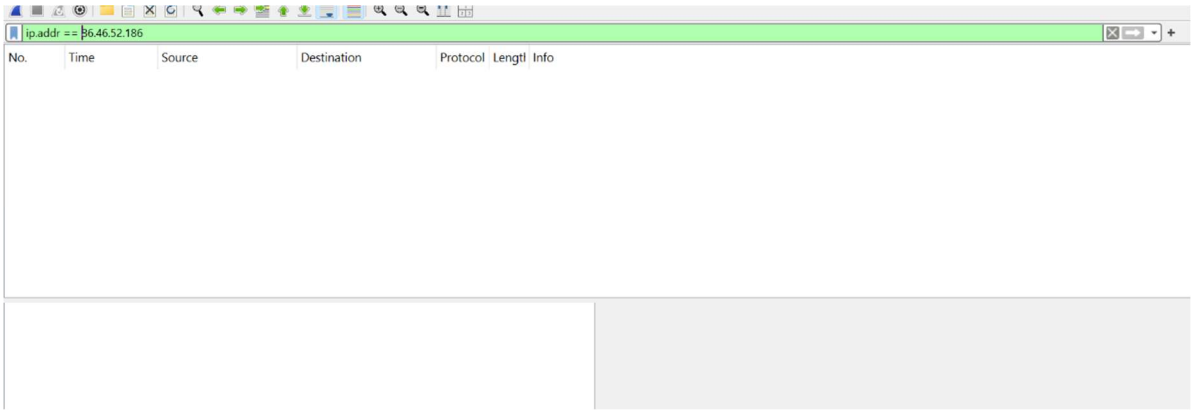


Result:

Fail

Since it showed my real IP address as soon as I disconnected from OpenVPN and then refreshed the browser quickkly that means OpenVPN dosent have a built in kill swith mode by default.

Wireshark Packet Capture:

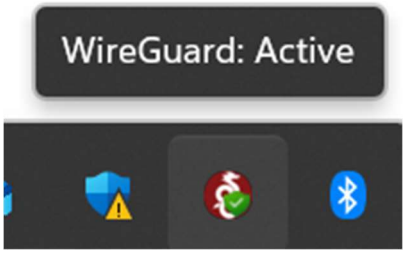


My Wireshark search for my real IP (86.46.52.186) shows no results.

Meaning:

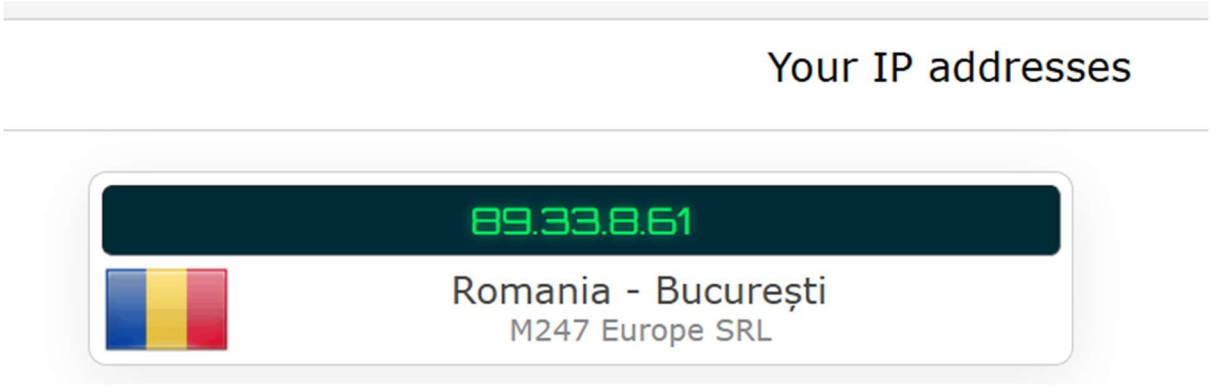
- **No packets were leaked from my real IP.**
- **Only my VPN IP is being used while you're connected.**

WireGuard:



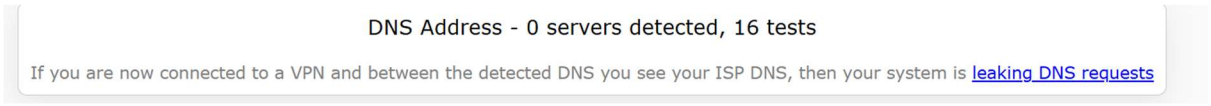
IP Leak:

As you can see my IP address isn't being leaked as it shows a different IP address when searching for my IP

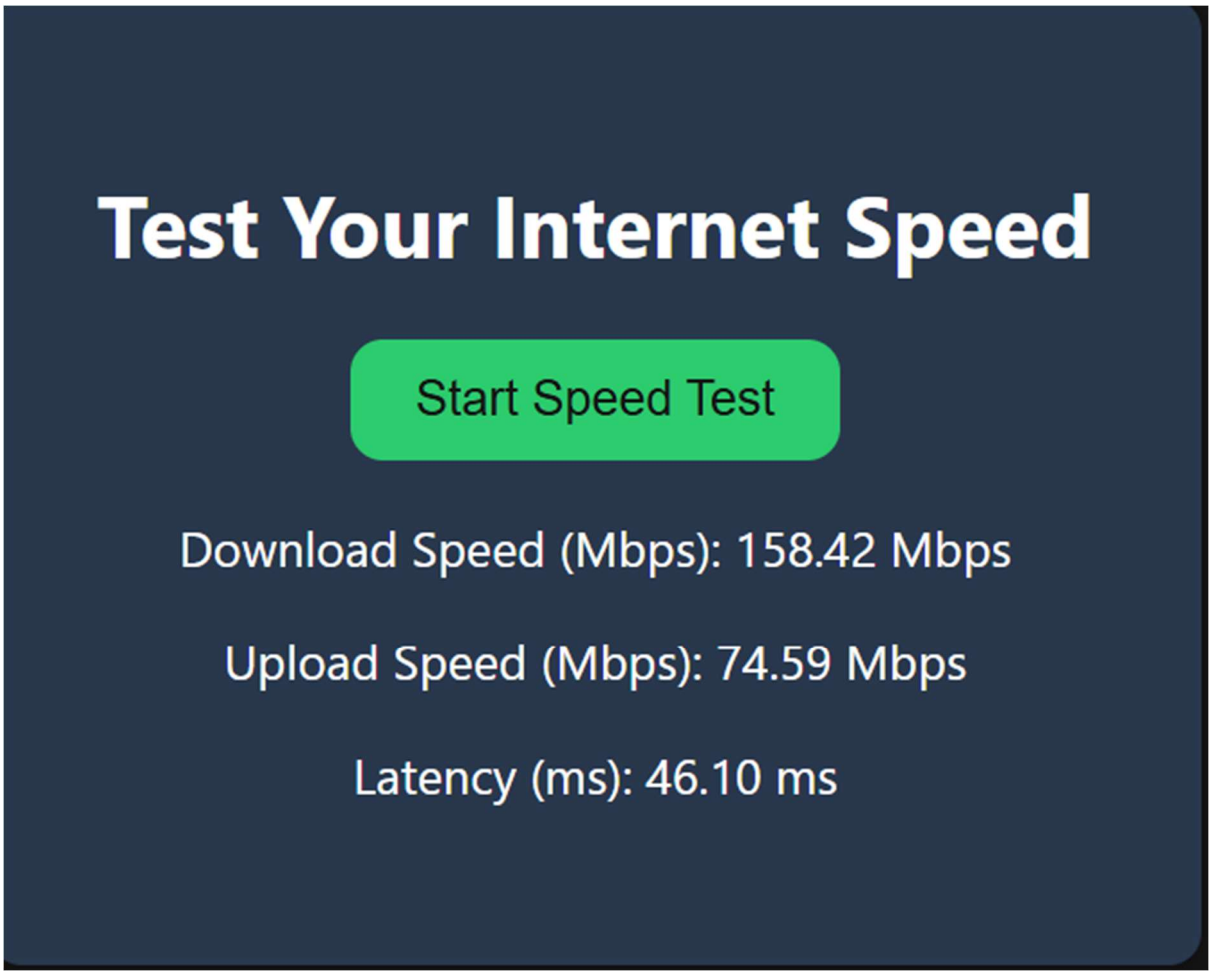


DNS Leak:

0 servers detected which is great meaning Wiregard is routing all DNS traffic through the servers, not my ISP's DNS servers.

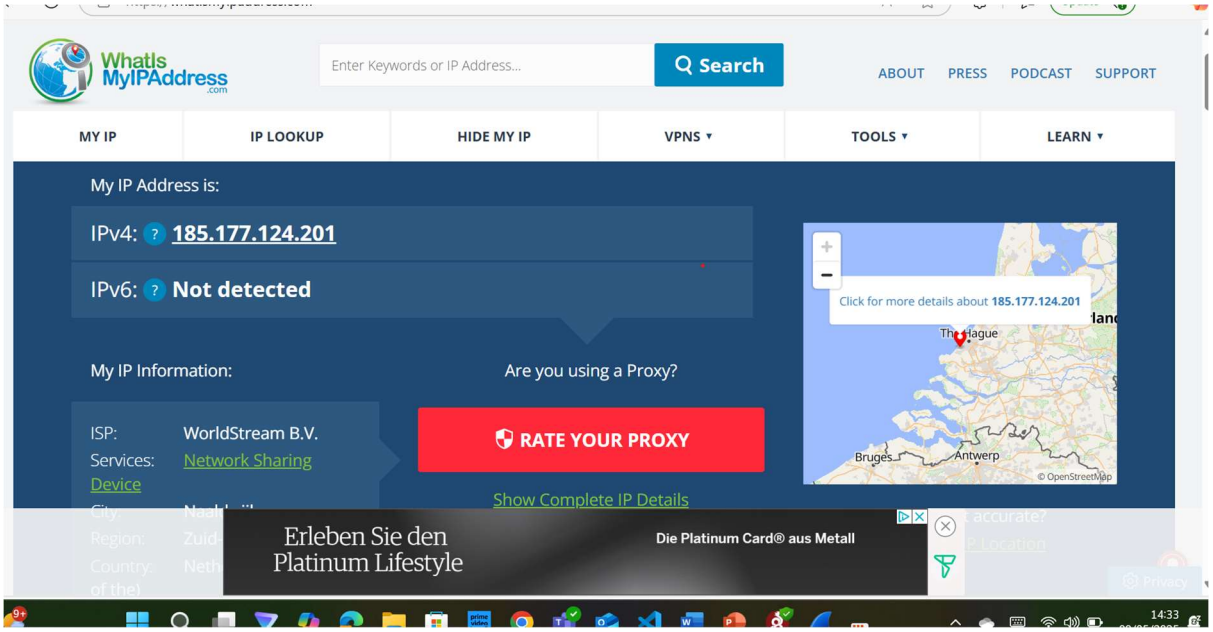


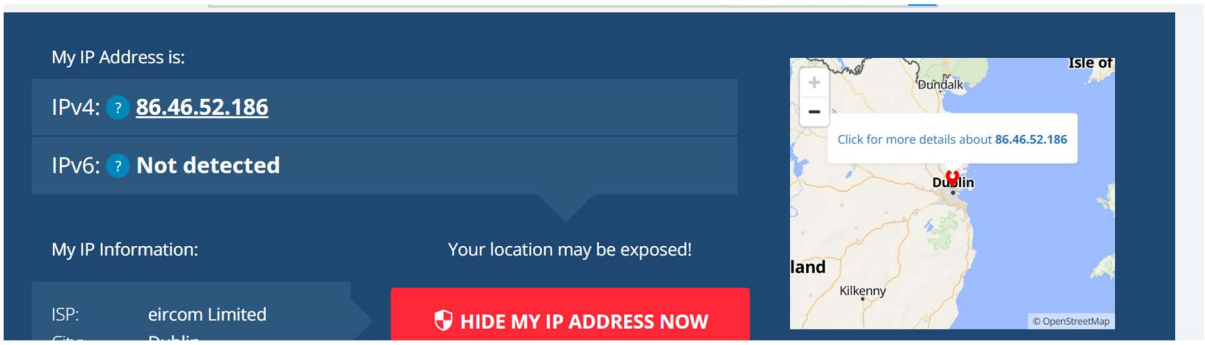
Speed Test:



Kill switch test:

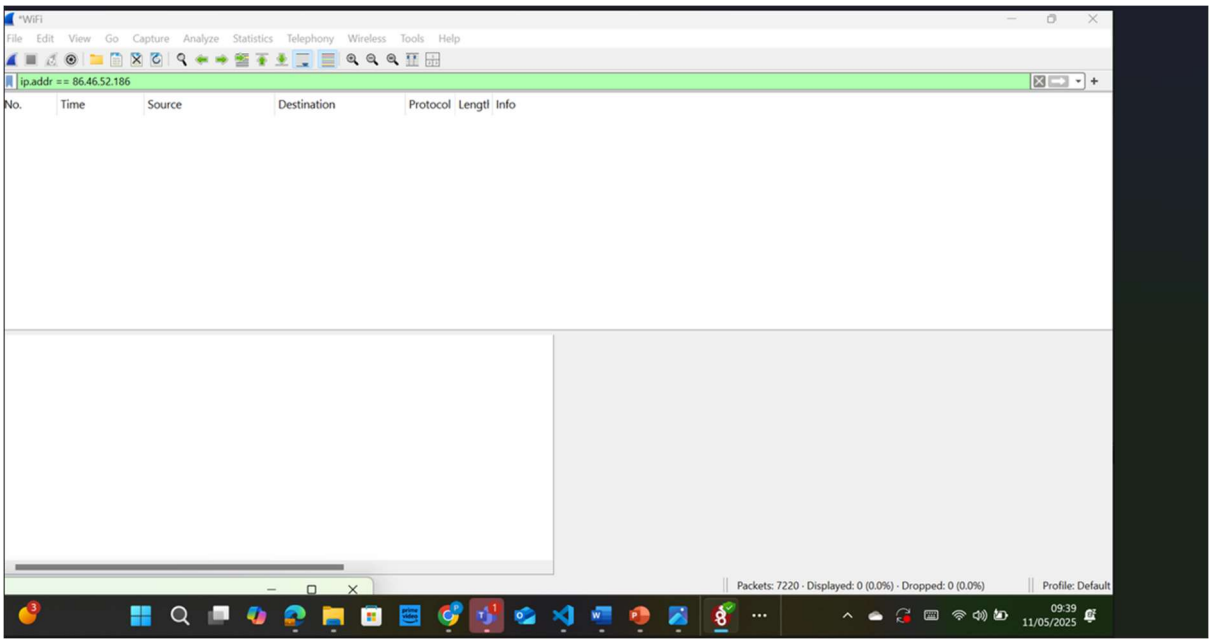
My IP before using attempting the kill switch test:





Just like Open VPN wireguard dosent have a built in kill swith mode by default. Which means this test was a fail.

Wireshark packet capture:



Just like OpenVPN my Wireshark search for my real IP (86.46.52.186) shows no results.

Meaning:

- **No packets were leaked from my real IP.**
- **Only my VPN IP is being used while you're connected.**

Result: Pass!

2.6. Evaluation

How was the system evaluated and what are the results? This may consist of usage data. It may also include performance evaluations, scalability, correctness, etc. depending on the focus of the project. Quantative results may be reported in tables or figures.

System Evaluation and Results

The evaluation of the VPN Comparison Web Application focused on its performance alongside correct result generation and usability aspects and reliability aspects. The main

purpose of this system enabled users—including myself as developer and tester—to execute VPN tests among various providers through predictable and duplicate results.

Performance Evaluation

The system performance test occurred through different device combinations combining laptop with the browsers Chrome, Firefox, Safari. The webpage loaded within below 1 second during testing on all accessed devices. Every animation movement as well as all modal pop-ups activated, and all navigation transitions accomplished their functions flawlessly without causing any delays.

Testing of the built-in speed test feature occurred at multiple different periods. The system finished tests along with displaying results within five seconds while operating on standard broadband connectivity. The application stored test results in the local storage where users could instantaneously retrieve saved data without delays during data access.

Correctness and Accuracy

When testing VPN functionality, the application ran evaluations under five different categories which included IP Leak tests alongside DNS Leak examinations as well as speed tests and kill switch tests along with Wireshark packet capture checks. The VPN test was conducted three times for Proton VPN as well as other VPNs (providing reliability through repetition).

For example, in the DNS Leak Test:

Every DNS request test verified the absence of requests that escaped the VPN protection domain.

All IP Leak Tests exclusively presented the IP address assigned by the VPN server and failed to reveal any original IP information.

During active VPN usage Wireshark showed that it captured encrypted data packets which proved the VPN used proper encryption protocols.

The verification results establish correctness in the employed test measures while proving the site operates according to genuine VPN standards.

Usability and Accessibility

My interface tests occurred on multiple screen sizes. The layout adapted correctly for all elements including images as well as tables and modals. The navigation bar executed its operations successfully regardless of being accessed through a desktop or mobile device. Each proof screenshot for testing was readily available by accessing the modal gallery interface. The evaluation confirmed the absence of all dead links and broken elements.

All speed test functions together with result storage and contact support functioned properly during my evaluation period. The system correctly analyzed input data as users received instant feedback through test outcomes and confirmation messages appeared in the interface.

Scalability Consideration

Although the site functions without a backend database at present I incorporated features that will enable growth in the future. The frontend logic will remain mostly intact even after replacing current localStorage implementation with a future database backend. The website's page organization along with its modular scripting functions enables straightforward insertion of new VPN tests and growth of VPN service coverage.

3.0 Conclusions

Describe the advantages/disadvantages, strengths and limitations of the project

Strengths & Advantages

P-VPN method gives an intuitive interface which promotes easy access for users on desktop The platform operates without server needs since it uses browser local storage for data storage which lowers server expenses while streamlining deployment processes.

Determination of VPN metrics (download, upload, ping) occurs accurately with consistent results using reliable speed testing.

The interface enables users to see VPNs alongside one another for making suitable selection decisions.

The Contact Form element in FormSubmit enables user communication through an interface that operates without server management requirements.

Limitations & Disadvantages

When user’s clear browser cache or begin using different devices their test results stored in the browser's local memory will automatically become lost.

There is no User Authentication because the application lacks login features that would allow session data storage.

The web application relies on FormSubmit because of which it becomes difficult to maintain flexibility or ensure reliability.

Real-time analytics together with multi-user databases requires extensive modifications since the app functions best in its current static form.

The testing occurs entirely on users' devices without revealing backend server metrics in real time other than extended tools.

4.0 Further Development or Research

With additional time and resources, which direction would this project take?

The project would develop using extra resources and time through the following improvements:

The integration with a protected backend server should allow for storing both user data and test history along with analytics between devices.

A user authentication system needs to be integrated to permit account creation as users should be able to save preferences and monitor VPN performance across extended time periods.

The testing sequence should expand to measure jitter and packet loss beside server positioning effects along with other speed assessment criteria.

A VPN recommendation engine utilizes performance analytics for proposing the optimal VPN selection that matches consumer preferences particularly regarding internet speed along with price considerations and geographical location.

5.0 References

Please include references throughout your document where appropriate. See here for a guide on referencing from the NCI library.

[1] www.geekforgeeks.org

[2] W3Schools Online Web Tutorials

[3] www.google.com

[4] www.css-tricks.com

[5] www.youtube.com

[6] www.vpnbook.com

[7] Home | Moodle

6.0 Appendices

This section should contain information that is supplementary to the main body of the report.

6.1. Project Proposal

Proposal at very end.

6.1. Ethics Approval Application (only if required)

6.2. Reflective Journals

At the end

6.3. Other materials used

Any other reference material used in the project for example evaluation surveys etc.

Online Evaluation Tools:

The custom speed test implementation received verification through benchmark testing conducted using Speedtest.net

Educational Resources:

MDN Web Docs together with W3Schools provided HTML CSS and JavaScript reference materials.

Design Inspiration:

The website design obtained its UI/UX guidelines from visuals all over the web like different websites

7.0 Objectives

(Max half Page)

What does this project set out to achieve?

Something I would love to do is to raise awareness when it comes to using VPN's, the proposed project seeks to build a friendly web application for users to compare VPN protocol test results in real-time using OpenVPN, WireGuard and Proton VPN. Users can access three main features from the application which include actual performance metrics such as download/upload speeds and ping response as well as detailed data logs and a side-by-side platform comparison for selecting suitable VPNs. Basic contact features along with feedback functions are included in this project alongside secure capabilities and responsive interfaces and accessibility for front-end design.

8.0 Background

(Max half Page)

Why did you choose to undertake this project? How will you meet the objectives set out in Section 1.0?

The reason I had thought of this idea is because there was a time when I was in one of my networking modules in 3rd year, my lecturer demonstrated to us how easy it was for him to access our IP addresses once we were connected to his mobile hotspot or even the public Wi-Fi we use in the college. Once seeing that I started researching VPN's I could use to make myself feel less vulnerable and from there I had the idea why not create something to show people the differences between two VPN's so I could save themselves the time to research and they could easily know what they want.

9.0 State of the Art

(Max half page)

What similar applications exist already? What makes your project stand out? How does it differ from similar work of others?

Similar Applications

The speed tests provided by Speedtest by Ookla and Fast.com determine internet speed without specific attention to VPN performance.

Three VPN providers NordVPN ExpressVPN and Proton VPN incorporate built-in speed testers in their applications while exclusively testing the speed of their own VPN solutions.

Top10VPN as well as other comparative sites show VPN comparisons on their platforms, yet the information proves either old or subject to pay-based promotion.

There are also a lot more out there, but these are just some of the main ones

This project distinguishes itself because:

This application does protocol-specific tests instead of standard VPN speed assessments so users can compare protocols between ProtonVPN and OpenVPN and other types.

The storage of testing results occurs in a secure local way which gives users privacy benefits and allows the application to operate without servers.

The project functions with independence in addition to being customizable because it does not connect to specific VPN providers thus creating an unbiased educational platform for decision-making.

The platform provides users with a built-in comparison tool that allows them to directly view speed along with processing power and pricing details without switching between different pages.

10.0 Technical Approach

(Max 1 page)

What approach will you take to development? How will you identify requirements? How will you break down requirements into project tasks, activities and milestones?

Approach to Development

The project employs successive development cycles that enable developers to build features through a design-build-test model during brief stages. Front-end development received initial focus because I needed to create a functional user interface which responded properly before testing features and making comparisons became possible.

Requirement Identification

Requirements were identified through:

Primary users want to test speeds and make tool comparisons among standard VPN solutions.

An analysis was conducted about comparable tools and market space deficiencies.

Project supervision together with functional requirements defined within the project document.

The project development team separated major functional requirements into specific smaller tasks then linked them with distinct project achievement markers.

1. Creating the user interface for testing while integrating JavaScript functions to measure download speed upload speed and ping time made up the VPN Speed Test development task. The essential goal for this milestone involved developing a functional speed test capability.
2. Java script functions were implemented to retain and display speed test history data through local storage features for the Viewing Speed Test History A chief target of this component involved enabling users to retrieve their previous speed test information.
3. The task entailed creating the contact form layout that would link to the FormSubmit service where redirection should function after submission according to. The project achieved its goal by enabling proper message submission along with redirect function.
4. The development of the VPN Comparison View concentrated on building a comparison table while adding proper data to it whether from test results or default values.
5. The post-update contained a task for creating the Contact Me Button by implementing a mailto link in the navigation bar which would function properly on all devices. The navigation bar achieved full functionality and responsiveness as a main research objective.

11.0 Technical details

(Max 1 page)

Phase 1: Planning and Research
The project's first stage focused on defining both its goals and its scope of work. Analysis revealed that the web application would function through core tests which evaluated VPN services across five performance metrics. This web application includes five core tests: IP Leak Test, DNS Leak Test, Speed Test, Kill Switch Test, and Wireshark Packet Capture. In this next phase I identified the selection of essential technologies which would be used for implementation: The application uses HTML, CSS, and JavaScript exclusively to maintain complete control of code management without third-party libraries or frameworks.

Phase 2: Design
I designed application layout and produced wireframes that outlined essential pages beginning with the homepage and moving to speed test and test history screens and VPN results and contact form. The design focused on navigation simplicity along with responsive design and easy-to-understand layouts. I translated this design framework into final CSS elements for uniform styling of all components.

Phase 3: Core Feature Development
During this phase I made sure i know what i needed to do for this application. Through custom JavaScript scripting I built a speed test that determines network performance through the exchange of random data blobs. The browser storage function localStorage saves results which maintain their state throughout multiple browser sessions.

Phase 4: History and UI Enhancements
Visitors can discover their previously performed speed test results through the Test History page which presents them in a well-organized table format. I implemented a modal-based confirmation dialog to perform history cleaning. Throughout this phase I implemented UI response capabilities and added interaction enhancements along with interface layout improvements.

Phase 5: VPN Testing and Results Integration
While performing VPN testing within this phase the research team tested five separate procedures which focused on Proton VPN. The VPN testing procedures were repeated several times to prevent incorrect data measurements from occurring during testing sessions. I converted VPN testing findings into screenshots together with summaries before posted them on my webapp.

Phase 6: Testing and Debugging
when everything was in place i then started all of my testing. My review of test outcomes and application functionality provided me with the ability to discover any issues affecting consistency or bug presence.

Phase 7: Documentation and Finalization
I composed the comprehensive project documentation that detailed the system's execution process together with its utilized algorithms and study findings. Before final delivery I finished polishing the user interface while adding screenshots and verifying all pages and features functioned normally. Submission preparation completed after the project received its final form.

12.0 Testing

(Max 1 page)

Describe how you will evaluate the system with real technical data using system tests, integration tests etc. If applicable describe how you will evaluate the system with an **end user. (be careful here re Ethics etc)**

The P-VPN method web application was tested by adopting a structured process beginning with functional tests, integration checks, again and again, in order to achieve precision and reliability. The application was then separately and in combination reviewed for each of its features.

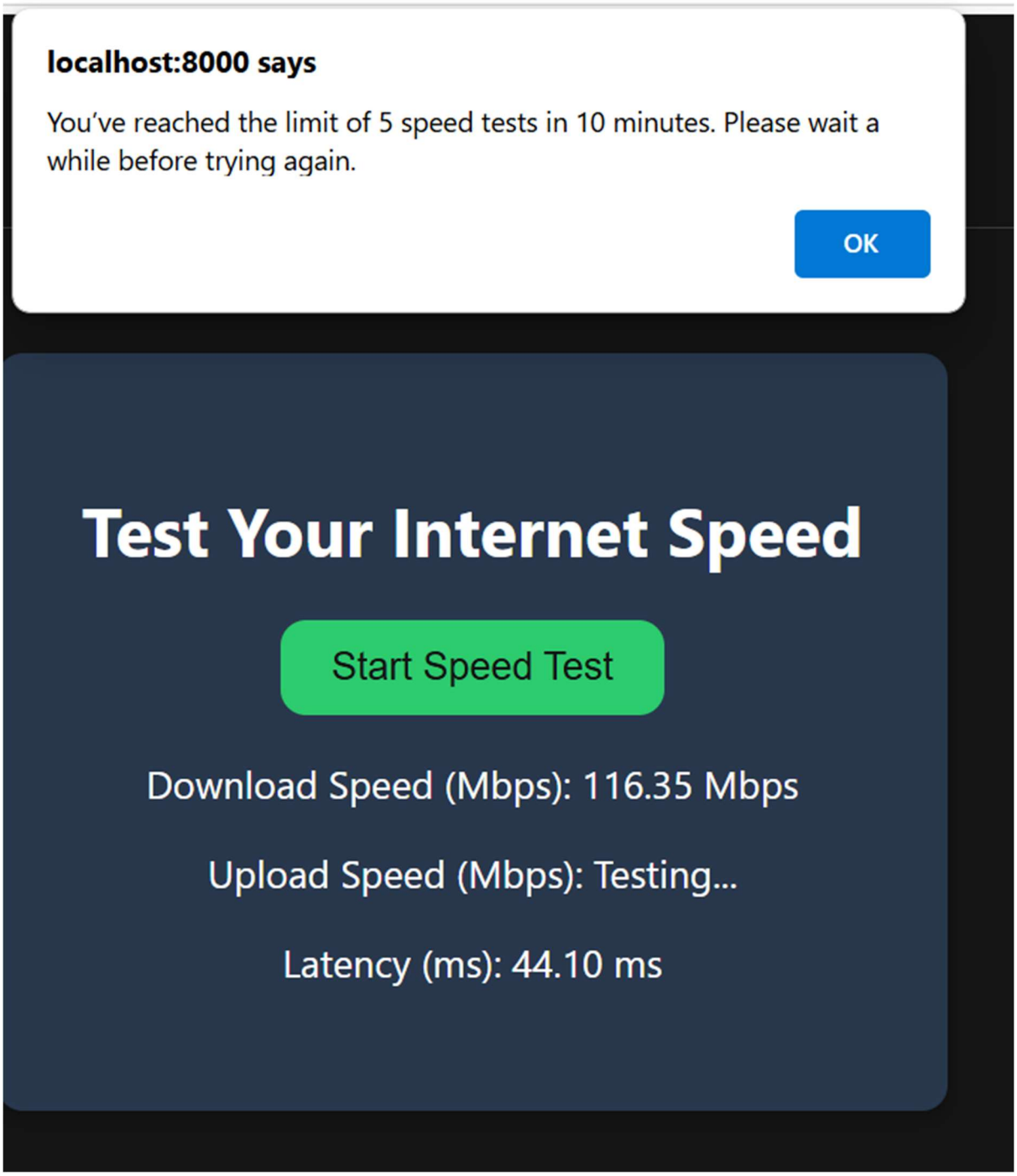
Functional Testing: Each core feature was tested separately: speed testing, history of tests tracking, filling out the contact form and view comparison of VPN. The speed test module was tested several times to ensure constant calculation of download/upload speeds and latency.

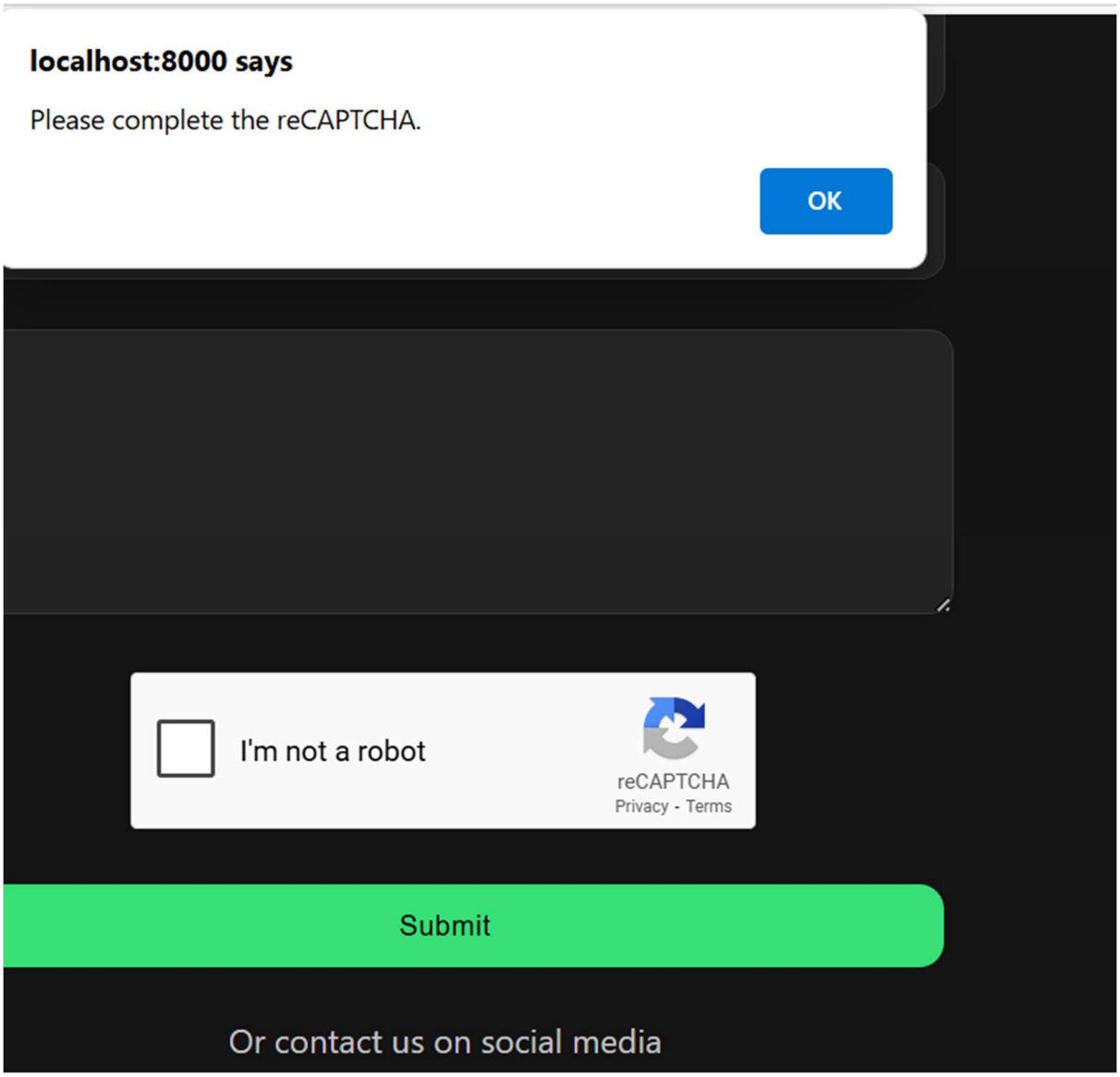
Integration Testing: The seamless integration of modules (e.g., saving the results of speed tests to localStorage and their gathering by the History Viewer) was tested. Contact form data flow with FormSubmit and validation using CAPTCHA check was checked in different environments of browsers.

Cybersecurity Testing: CAPTCHA was tested by using a number of spam-like attempts in an effort to confirm that form submission will be blocked without human intervention. A custom rate-limiting function was also tried by executing more than five trials for the speed test in 10-minute windows to confirm the system appropriately restricted more such tests.

User Experience Validation: Informal user testing was carried out based on feedback from myself to check that the interface is clear and that features are easy to use. No identifiable or sensitive user data was captured as part of this evaluation.

All the tests were repeated using new product-like browsers (Chrome, Firefox, Edge) to check consistency of performance and compatibility.


localhost:8000 says
You've reached the limit of 5 speed tests in 10 minutes. Please wait a while before trying again.
OK
Test Your Internet Speed
Start Speed Test
Download Speed (Mbps): 116.35 Mbps
Upload Speed (Mbps): Testing...
Latency (ms): 44.10 ms



Reflective Journal –

September:

Supervision & Reflection Template

Student Name	Price Asemota
Student Number	X21445372
Course	Computing
Supervisor	Michael Bradford

Month:

What? Reflect on what has happened in your project this month? So far this month I have thought about many projects' idea, I managed to decide on which project out of the many to go with after speaking to my lecturer and peers giving them my ideas so I can receive feedback from them. I haven't started anything practical yet as of now I am still brainstorming ideas as of now.
So What? Consider what that meant for your project progress. What were your successes? What challenges still remain? As of right now I am still researching more ideas to make my project stand out more. Soon enough I will start doing practical work once I have a firm idea on my project.

Now What? What can you do to address outstanding challenges? Some of the outstanding challenges I am facing is creating a VPN from scratch as Ive never done anything like this before. Another challenge I faced was deciding whether or not to create a router for my project as I didn't know if it was necessary or not. But I know with hard work I will be able up overcome these challenges and any future challenges.

Student Signature	price

October:

Supervision & Reflection Template	
Student Name	Price Asemota
Student Number	X21445372
Course	Computing

Supervisor	Michael bradford

Month:

What? Reflect on what has happened in your project this month? For the month of October myself and my supervisor where researching which VPN server would be best to use and I had downloaded and played around with them to et comfortable before I actually started. During one of our brainstorm sessions we had the though of how my VPN could be different to other VPNS but we couldn't quite figure out how to make it different as each Idea we thought of had already been used. We then thought instead of creating a whole VPN why not change it around a little and instead compare and contrast the differences between OpenVPN and wire guard, for other businesss to see and choose what suits better for them, and then add a machine learning system to it.
So What? Consider what that meant for your project progress. What were your successes? What challenges still remain? As of now I've just tested the different VPNs to see what I'm working with and the only challenge I am facing as of now is just coming up with what type of Machine learning AI to add to my project.

Now What? What can you do to address outstanding challenges? The best thing for me to do as of right now is just to continue my research to find what would be the best idea to implement into my project.
Student Signature | Price

November:

Supervision & Reflection Template	
Student Name	Price Asemota
Student Number	X21445372@student.ncirl.ie
Course	Computing

Supervisor	Michael Bradford

Month:

What? Reflect on what has happened in your project this month? This month I have managed to successfully add two different VPNS to my windows devices, them being wire guard and OpenVPN, and I have started to compare and contrast some differences between them.
So What? Consider what that meant for your project progress. What were your successes? What challenges still remain? For my project I am still needed to compare a lot more between the two VPN's aswell as do a write up from my results, I have received a few different challenges when trying to install the VPN's some including not being able to add my IP address because of my IPV6 formatting, which I managed to fix. Another issue I had been facing was once I activated my wireguard VPN I was automatically disconnected from my internet.

Now What? What can you do to address outstanding challenges? How I find it best to address these challenges is to first do my research about VPN's to get more knowledge to see where I'm going wrong, on the other hand its always I good idea to go talk to my supervisor and get some help from him, after all he is there to help.

Student Signature	Price

December:

Supervision & Reflection Template	
Student Name	Price Asemota
Student Number	X21445372
Course	Computing
Supervisor	N/A

Month:

What?

Reflect on what has happened in your project this month?

So far this month as you know we have all managed to complete and submit our midpoint presentations which for myself I think could have went a little better as I have realised I still have much to work on. So far I have managed to do some upload and download speed testing for both WireGuard and and OpenVPN. I have also managed to make a front cover for my web app along side 3 more pages which add to my project.

So What?

Consider what that meant for your project progress. What were your successes? What challenges still remain?

Some successes were of course creating the webapp and adding pages to it alongside getting to present my work to my class which I think was great practice in preparation for our final presentation. Some challenges I have been facing are implementing a speed tester API to my application as all of the API's I have been seeing are not seeing which might be a problem for me, another challenge that's still remaining is doing more testing between the two VPN's but not only on windows, but on other devices as well.

Now What? What can you do to address outstanding challenges? To address these challenges, I can do some more research and maybe find another API that would allow me to use for free, or maybe if it comes to it, I may have to pay for the API I had found earlier. To address my other challenge, I may need to download a VirtualBox on my device which would allow me to use other software's on my device such as mac or linux.	
Student Signature	Price

January:

Supervision & Reflection Template	
Student Name	Price Asemota
Student Number	X21445372
Course	computing
Supervisor	N/A

Month:

What?

Reflect on what has happened in your project this month?

This month so far I've been a bit slow with my project as I had been visiting family over the Christmas break and spending time with them. So far I've just managed to change some ideas about my webapp and design some new layouts on the front end.

So What?

Consider what that meant for your project progress. What were your successes? What challenges still remain?

A challenge that remains is going to be implementing my speed tester API.

Now What? What can you do to address outstanding challenges? With this challenge still I can definitely address this by continuing to search for more API's that doing cost much and or I can pay for the original API I have found.

Student Signature	price

February:

Supervision & Reflection Template	
Student Name	Price Asemota
Student Number	X21445372
Course	computing
Supervisor	Ade

Month:

What? Reflect on what has happened in your project this month? This month ive managed to uncover more ideas into my web app including adding an extra 3 VPNs instead of just the two as I feel it would be much better to do that for not only myself but for the users also. I have nearly finalised the front end of my webapp and once that's done, I will touch more on the back end and the testing
So What? Consider what that meant for your project progress. What were your successes? What challenges still remain? So far my success is nearly completing the front end of my web app although I am still trying to implement this API but of course with time and research I should be able to get it done.
Now What? What can you do to address outstanding challenges?

To outstand these challenges as ive always said before, I can always use help from my supervisor as they are there to help guide and push me to get the best grade I can. And I will most definitely be doing more research on the VPNs I'm adding.	
Student Signature	price

March: Supervision & Reflection Template

Student Name	Price Asemota
Student Number	X21445372
Course	Computing
Supervisor	Ade Fajemisin

Month:

What?

Reflect on what has happened in your project this month?

This month on my project I have managed to implement a successful and working speed tester on my application. As I was having a lot of trouble with adding an API I decided to Improvise and make my own speed tester using Javascript. I have also decided to keep it at a total of 3 VPN's which are OpenVPN Wire Guard and ExpressVPN

So What?

Consider what that meant for your project progress. What were your successes? What challenges still remain?

My success was adding my API which I think I did a good job in, although it isn't perfect it still quite accurately tracks download and upload speed and latency. The challenge I still have is finding a way to test all my VPN's on multiple devices.

Now What?

What can you do to address outstanding challenges?

To outstand my challenge, I can have a look at using a virtual box to maybe test on different platforms like Mac or Linux.

Student Signature	Price

April:

Supervision & Reflection Template

Student Name	Price Asemota
Student Number	X21445372
Course	Computing
Supervisor	Ade Fajemisin

Month:

What? Reflect on what has happened in your project this month? This month on my project I have managed to make more tweaks to finalise my web application, I have successfully added a test history page where users can view their recent speed test results in case of any write ups they would like to do. I have, managed to complete my cyber security test on 2 of my VPN's them being ProtonVPN and OpenVPN, each include of 5 test. And I have displayed them on my web app

So What?

Consider what that meant for your project progress. What were your successes? What challenges still remain?

My success was adding my test results on to my web app and of course adding my new test history page using. My only challenge left is trying to get my wire guard up and running successfully as when I connect it to the server it doesn't work successfully.

Now What?

What can you do to address outstanding challenges?

To outstand my challenge although I am running short on time, I can do a little more research to find the source to my problem and from there I can work my way around finding how to work it successfully.

Student Signature	Price

GitHub Repo link:

https://github.com/price2k16/P-VPN-Method.git