**ISRAEL ADEYEYE**   
**2423121007**  
**CA-2 REPORT**

**1. Language Used and Website Functionality**

In this project, I used HTML, CSS, and JavaScript to create a web-based application that allows users to check in at a specific location using geolocation. The website is designed for two types of users: admins and regular users. Admins can set a workspace location (latitude, longitude, radius, and vertical tolerance) and manage access codes, while users can enter their details and check in only if they are within the defined area. The code uses the Geolocation API to fetch the user's current location and calculates distances using the Haversine formula to determine if the user is inside the allowed range. For example, if a user is more than 500 meters away from the workspace or outside the vertical tolerance, they cannot check in. This was my first time working with APIs like Geolocation, and it felt empowering to see how simple code could interact with real-world data like GPS coordinates. I also learned how to store and retrieve data using localStorage, which helped me understand how websites can "remember" user settings without needing a backend database.

**2. Digital Transformation and New Tech/API Tools Discovered**

This project opened my eyes to the power of digital transformation and how technology can replace traditional methods of monitoring attendance or access control. Instead of manually signing in or using expensive subscription-based apps, I now know how to build a custom solution tailored to specific needs. One of the most exciting discoveries was learning about APIs—tools that allow developers to tap into external services or hardware features. For instance, the Geolocation API made it possible to pinpoint a user’s exact location, and I realized how widely applicable this is for applications like delivery tracking, fitness apps, or even augmented reality games. Additionally, I explored concepts like responsive design and browser compatibility, ensuring the app works well on different devices. Another game-changer was discovering AI tools like GitHub Copilot and ChatGPT, which helped me debug errors faster and brainstorm ideas when I got stuck. These tools showed me how AI can enhance productivity and creativity in coding projects.

**3. Things I Learned**

One major takeaway is that many paid applications we use daily—like attendance trackers or location-based services—can actually be built by individuals with basic programming skills. Knowing this gives me a sense of independence because I no longer feel reliant on third-party subscriptions. I also learned how to break down complex problems into smaller steps, such as calculating distances between two points on Earth’s surface or validating inputs securely. Most importantly, I saw firsthand how AI enhances the development process. Whether it was generating boilerplate code, explaining tricky concepts, or suggesting improvements, AI significantly sped up my workflow. This combination of technical knowledge and modern tools has inspired me to continue exploring ways to solve real-world challenges through coding. It’s amazing to think that just a few weeks ago, I didn’t know where to start, and now I’ve created a fully functional web app! I now understand how to create a custom solution tailored specifically to my needs using HTML, CSS, and JavaScript. This newfound independence has been incredibly empowering.

However, the journey wasn’t without its challenges. At first, I struggled with understanding how to integrate real-world data like GPS coordinates into the app. For example, when working with latitude, longitude, and altitude, I initially worried about inaccuracies due to factors like Earth’s rotation or poor GPS signals indoors. While researching, I learned that these concerns were valid but manageable. The Haversine formula, which calculates distances between two points on a sphere, helped address horizontal positioning, and adding a vertical tolerance range made the system more forgiving for altitude discrepancies. Another challenge was ensuring the app worked seamlessly across different devices and browsers. I had to test extensively to ensure compatibility and handle edge cases, such as when geolocation data wasn’t available or when users entered invalid inputs.

AI tools played a crucial role in overcoming these challenges. When I encountered issues, such as debugging errors or figuring out how to implement specific features (like auto-setting the admin's location), AI assistants provided clear explanations and code snippets that saved hours of trial and error. For instance, when I complained about the complexity of accounting for altitude, AI suggested simplifying the process by introducing a vertical tolerance instead of requiring exact values. This small tweak made the app much more user-friendly and practical. Additionally, AI helped me grasp advanced concepts like API integration and responsive design, which would have taken much longer to learn on my own.

This experience has truly transformed my perspective on what’s possible with technology and inspired me to keep learning and experimenting.