“Why did I think I could do this? I should have stuck with what I knew” I said to myself sitting in the university library at the end of the first semester, second year. About a year ago, a friend of mine was working on his blog when I noticed his mumbling while I was sitting at this very table. After watching him and being confused for a while I started understanding what each block (i.e. HTML tag) was doing and got interested. The lights shut off and the librarian yelled “we are closing guys”, it was 10 pm and it’s been hours since the last time I took my eyes off the screen. Coming up with an idea and slowly block by block turning it into reality, this immersive experience reminded me of something. It was in my childhood - playing legos. In the coming weeks, I built my own blog and a number of other websites, everyday tinkering with HTML and CSS became my hobby. At the end of the semester, I looked up and BIS was the only major that offered courses that taught building web, desktop, and mobile applications. My long time goal of becoming an economist like my father was out of the window by then.

I had chosen BIS following my curiosity in building websites. However, I was still learning and knew little about programming, so I had taken a number of programming courses. Few weeks into the studies I found myself struggling in an increasingly unfamiliar field of study, algorithms, data structures, and programming languages. Additionally, after part-time work, I found little time and energy left to wrestle with these new content while being overwhelmed to do programming in a complex environment such as Visual Studio. As a result, the first semester was less than expected. One of my TA’s come up to me when I was sitting in the library disappointed. Upon listening to my story, she suggested learning python preferably on a command line. I turned my full focus to learning programming and decided to take a break from work and all other activities. I would watch python tutorial videos while eating lunch and fall asleep while implementing merge sort. Programming and problem solving become all I could think of when I joined an online programming community. I easily picked up C#, Java, and later other programming languages once I was competent in Python. Turning my grades in the second semester and getting the highest mark, 99 from the second programming assignment (Fundamentals of Programming) was the result of all that work. This experience had become an example of hard work and resilience that motivated me throughout my studies and my work.

How I viewed software development came to take a different shape when I took *Software Analysis and Design* course. “The code is the truth, but not the whole truth” the professor quoted Grady Boosh and said, “I will teach you the rest of it”. Sitting down and writing code - that is how I viewed software development at the time. I learned that though it’s part of the process, building software that is intuitive and that stands against the test of time and that can support one or million users, requires a lot more than just writing code. It was a survey course that introduced topics including software architecture, development lifecycles, requirements engineering, UML modeling, human interface design, and others. Since then I have been independently exploring these topics which lead to my decision to pursue a master's degree in computer science.

Starting from summer, after the third year, I started interning then later working as a software developer at Datasite, a local IT company where I had a number of experiences that reinforced my decision to pursue further education. Developing GoTest, a university entrance exam simulation system was one of them. I developed a RESTful web service and participated in the design and analysis of user experience. While developing the web service I was able to accomplish, through various optimization techniques including implementing the majority of the data manipulation in SQL stored procedures, in-memory caching to cache heavily computed responses, to achieve almost instantaneous response time. Although we had followed all the guidelines that we knew starting from researching the users (i.e. students preparing for the exam), establishing user pathways and objectives, developing a prototype to iteratively testing and adjusting through user feedback and various usability metrics, we were outdone by another company. Following our release that company, ABT has also released a similar product. In spite of having fewer features, in two months ABT had fifty thousand users while we had only two. Through reading user feedback on distribution platforms we had discovered that competitor's solution had a familiar look and feel. They used a number of user interface elements including outline, spacing, font type, and other elements similar to official tests with which the students grew familiar through months of preparations. Our competitor communicated with users through a familiar language while we created a new one, a better one we believed. They had created an extension of the user's experience in the real world. Our UI might have been rated higher if judged based on usability metrics however competitor's solution was superior given the user's experience. This was one of my early experiences that illustrated the importance of understanding human user interaction and sparked my interest in the area. I believe the highly reputed research environment and courses including (Human-Computer Interaction Models, Theories, and Frameworks), and (Software Usability) make Carleton University the right place to explore my interests in the area.

Apart from university studies I have taken a number of online courses including "Algorithms and Data Structures" and "Software Engineering Essentials", formed Coding Club to help students, struggling with programming, participated, and won a number of online programming competitions. Upon my graduation, I became a software developer, and since then, I have built and published a number of native and cross-platform mobile applications. I have taken and successfully passed the C# (70-486) programming exam. Further, participated in two startup competitions and reached the semifinals in the second one, Covid-19 Challenge. In order to prepare myself for mathematically rigorous graduate studies, I have also taken the “Introduction to Math for Computer Science” course.

As a freshman, I had a plan to become an economist. But my curiosity led me to Computer Science. Currently, the same intuition is leading me to Carleton. Among other reasons, Carleton’s academic excellence, long-established research reputation in HCI, and diverse and welcoming atmosphere were the main factors to trust my education to Carleton. During my studies, I will be looking forward to taking (Computer Animation) and (Computer Graphics) courses and explore these areas in depth while complementing my existing knowledge and experience with (Mobile Commerce Technologies), (Advanced Database Systems) and other courses offered by Carleton. I am ready to make the best use of every opportunity offered by Carleton while contributing in any way I can.