Computer programming transcends mere technical skills; it is an art that breathes life into technological artifacts. This powerful tool empowers individuals to innovate and transform not only their lives but also their communities. Imagine young minds developing mobile applications to aid visually impaired peers in navigating school corridors or employing audio classifier technology to interpret an infant's cries, offering invaluable assistance to parents. These endeavors are more than mere projects; they are profound expressions of empowerment, reshaping perceptions of community and self. Yet, the journey to such innovative creations is often obstructed by the complexity of technology. Recognizing this barrier, my research is driven by a deep commitment to democratize computer technology. I aspire to pioneer approaches that simplify technological interfaces, making them more accessible and intuitive. My goal is to unlock a world where programming becomes a universal language of expression and innovation, available to all, irrespective of their background in technology.

In my career, I have pursued the field of educational technology. My journey reached a pivotal phase over the past two years through my collaboration with the esteemed MIT App Inventor team. App Inventor epitomizes my vision of technology democratization, enabling users from diverse backgrounds to create mobile applications without needing a foundation in tech. My contributions span pivotal areas such as conversational AI, data science, and the Internet of Things (IoT). A project that stands as a testament to my endeavors is Aptly. Aptly is a groundbreaking initiative under the generative programming umbrella seeking to revolutionize programming. Aptly allows users to build functional mobile applications using simple natural language. For instance, a user could state, “Make an app that translates English to any five languages” and Aptly would bring this idea to life as a fully functional app.

My journey with the development aspect of Aptly has been deeply fulfilling, sparking an interest in exploring its research dimensions. This shift in focus is what motivates my decision to pursue graduate studies. Aptly, as a code generation platform, stands at the frontier of a new era in programming, presenting intriguing implications and uncharted territories for future exploration. Within the realm of generative programming, two pivotal research questions have crystallized as the core of my academic curiosity:

1. What are the limitations of Generative AI in programming, and can we improve it further?
2. How do these AI-based no-code platforms change computer science education?

My first research question delves into advancing the technology of generative programming. This involves exploring and refining techniques like prompt engineering, fine-tuning, and potentially developing new deep-learning algorithms to enhance code generation efficiency. A critical aspect of this inquiry is assessing the extent of these models' understanding of programming languages and identifying their inherent limitations. A key consideration is whether these systems can comprehend the context of a program and accordingly update or generate code.

The second question involves how these platforms change the dynamics of programming and how that changes future computer science education. In its essence, programming can be seen as a form of conversation between a human and a computer. The programmer communicates their intentions to the computer in a language it can understand (i.e., a programming language), and the computer responds by executing the commands or providing feedback in the form of output or error messages. However, with the rise of platforms such as Aptly, this dynamic is starting to shift. These platforms allow users to create software via natural language. The underlying platform translates these natural languages into traditional code that the computer can understand. In this new paradigm, the conversation is becoming more human-centric.