Having never worked in the technology field before, security has often been a nonexistent thought in my personal work. I’ve always focused more on some other area of my projects, be it learning a new language, using new tools, implementing features, or just plain getting anything to work at all. Consequently, security has just not been a topic that I had done a lot of research on for these projects. As this course comes to an end, I am happy to now be able to say I know much so more about security, and how to start applying it.

One way to approach security is by adopting a security policy. A security policy is a document that establishes which coding standards are going to be followed and why, as well as explanations of potential threats and how to respond to them. Moreover, a security policy is invaluable for ensuring that every member of the team is on the same page regarding best practices. In fact, adopting a security policy at the beginning of the software development lifecycle is a great way to ensure that security needs are addressed early on in the design of the project. This can help ensure that security measures are being implemented at every step in the process, and that they are not simply being left as something to quickly add at the end.

There are always going to be potential threats to any system or application. Analyzing and understanding these risks are crucial to ensuring the security or stability of a project. A risk matrix is a great way to organize these threats in order of likelihood, severity, or potential consequence.  Doing this can allow an organization to determine where their resources are best used to keep their systems or applications secure. As an extension of this, defense-in-depth is another approach by which a system can be properly analyzed to determine what security measures should be taken to keep data safe.

Another meaningful approach to security is the zero-trust methodology – the idea that devices, users, transport/sessions, applications, and data should not ever be inherently trusted.  Requiring two-factor authentication is one example of not just simply trusting a device or user just because they’ve entered a few correct credentials. Operating under the principle of least privilege is another example, as it only allows authenticated users to access just enough to properly perform their job. Sanitizing user input is a way to prevent unintended input from causing harm or other unintended side effects on a system as well.

Security is an ever-changing landscape. As technology changes and evolves, so do the methods by which bad actors can threaten an application or system. A company cannot simply decide on a set of security policies once and never have to think about them again. It is crucial to stay aware of new vulnerabilities as they are reported and actively take steps to address security issues before they become a problem. This very well means that security rules and policies must constantly be revaluated and modified to fit both the needs of the company as well as security trends.