Security resources:

* [How to Implement Security in ASP Net Web Application​ - Tolu Michael](https://tolumichael.com/how-to-implement-security-in-asp-net-web-application/)
* [Secure Coding Guidelines for ASP.NET Core MVC & Web API](https://www.c-sharpcorner.com/article/secure-coding-guidelines-for-asp-net-core-mvc-web-api/)
* [Web API Security in .NET Core](https://www.c-sharpcorner.com/article/web-api-security-in-net-core/)

User account creation (& security):

* [Creating Users in ASP.NET Core Identity: A Step-by-Step Guide to User Registration](https://www.webdevtutor.net/blog/creating-users-aspnet-core-identity)

Key things to keep in mind:

* HTTPS is king – enforce HTTPS and SSL/TSL wherever possible to ensure secure communications
  + HSTS to keep communications only over the HTTPS server
* Role-based security & least privilege to prevent unauthorized edits/access
* Configure secure cookies using HTTPOnly or Samesite
* Parameter queries to prevent SQL injection
  + AntiXSS input libraries
* Encrypt stored data and configuration files
* Logs are good for reviewing any gaps in security – beneficial if applicable
* JSON tokens will be instrumental in establishing API security

To consider:

* Account security – won’t be storing sensitive data beyond email addresses, passwords, and maybe names
  + Email and name the only PII that will be requested
* Security for the chatbot will need to be configured to avoid injections in that aspect and other potentials vulnerabilities.