Basic web app security checklist

- [ ] make sure that all traffic is encrypted (using tls & https)
- [ ] hash all passwords in your database using bcrypt, so even if you get compromised it would take more than a lifetime with supercomputers to decrypt the passwords for logging in
- [ ] do not have the decryption key close to your sensitive data (e.g. both in the same database)
- [ ] rate limit actions (e.g. you should be restricted for some time if you have more than 5 login attempts per minute)
- [ ] check on server (& maybe client) if the user shall be able to perform a specific operation.
- [ ] log security breaches
- [ ] maybe allow only one session per user at a given time (login on another system -> loose login on another system)
- [ ] keep your software updated
- [ ] keep your systems updated
- [ ] in case you use 3rd party libraries, keep them updated regulary... you may not even notice that their code contains a security vulnerability. And it's not your code. Still it may be a major problem thats caused by even unused functions...

General Patching Strategy

- [ ] Keep an inventory of all services and applications you deploy
- [ ] Keep an inventory of all associated assets, libraries and softwares
- [ ] Standardize all production systems to use the same libraries, versions and operating systems - track the various update channels
- [ ] define the security controls you have in place (firewalls, rate limiting, tls, authentication mechanisms)
- [ ] have an update-plan and follow it!

Avoid using a third-party service or library because they can have uncontrollable risks

- [ ] Can you build the service or functionality yourself?
  - [ ] Never write your own cryptography!
- [ ] In case you can build it yourself, does it create more risk or less risk?
- [ ] Evaluate the packages you use! Check them for security flaws, tests, is it active?

Advanced

- [ ] add two factor authentication
- [ ] log every move of every user

- ☐ add snyk, sonar cube or something like that to your automated build pipeline