Security:

Mysql: We hardened the mysql server using mysql_secure_installation. This includes setting a strong randomly generated password for root, removing anonymous users, Disallowing root login remotely, and removing test databases,

Client and Server: The biggest thing here is to have the client encrypt data and then have the server decrypt it and store it on the local mysql database. However given the nature of our client and server we can encrypt to prevent from man in the middle attacks, but our code is exposed since it is javascript which means there isn't much we can do to hide our encryption methods to our users. I would like to do this using ssl/tls but it will be more practical to use the crypto module and make our own. MQTT doesn't have encryption built in so the data needs to be encrypted before it leaves the client and decrypted on arrival at the server.

Server Expressjs Webserver: We Installed the helmet module which provides protection against well known web vulnerabilities by setting the HTTP headers appropriately. Helmet does this by installing nine smaller middleware functions that set security-related HTTP headers. This includes csp, hidePoweredBy, hpkp, hsts, ieNoOpen, noCache, noSniff, frameguard, xssFilter.

Dependencies: Dependencies give nodejs a lot of power however they can also be a source of insecurity so I installed two vulnerability scanners nsp check and snyk. Nsp check will only check for vulnerabilities and give suggestions. Snyk will scan for vulnerabilities and then patch or update for you.