CST2572 Assessment 1 Self-Assessment Marking sheet, Information and checklist

Marking details:

Basic designs for both site and database (10%)

- *Front page designs (8%)*
- *Database E/R diagram(2%)*

IndexedDB database in use and managed (10%)

- *Creation of DB, with JSON data from server (5%)*
- *Add (2%)*
- *Delete (1%)*
- *Update (2%)*

Patient, Admin and Doctor account types and permissions (5%)

- *Patient account type and permissions (1%)*
- *Admin account type and permissions (2%)*
- *Doctor account type and permissions (2%)*

Admin functionality (Can manage patient's information name, address, DOB etc.) (10%)

- *Can add details of patient (name, address, telephone, DOB, NHS number (5%)*
- *Can update/delete (5%)*

Patient functionality (Can book appointments, look at medication instructions etc.) (10%)

- *Can book appointments and request prescriptions (5%)*
- *Can see stored information and medical notes (5%)*

Doctor functionality (Review patient history, medical notes, medicines) (10%)

- *Can look at patient history, notes (5%)*
- *Can add to patient history, notes and medicines (prescriptions) (5%)*

Validation checks and basic security (input sanitisation, password, email etc.) (5%)

- *Accounts are password protected (2%)*
- *Inputs are sanitised, password reaches applied security standard, email input is valid (3%)*

Security (Encryption) (10%)

- *Encryption applied to passwords and other data (5%)*
- *Decryption and data in use after storage (5%)*

Advanced (10%)

- *JSON data collected from server and processed (5%)*
- *Encryption modules used from crypto lib or similar, must be AES 256 or better (5%)*

Case Study / Report (20%)

- *Servers / Protocols*
- *VMs*
- *Hacking detection / mitigation*

Checklist:

(Code is marked anonymously, where possible, hence I just need your Student ID)

Student ID filled out on marksheet (below) ☒

Code included in zip ☒

Code is commented ☒

Code is modular (e.g. broken into functions) ☒

Readme.txt is supplied giving references, usage etc. ☒

Video included in zip ☒

Video time: 4 minutes

Video shows all claimed sections above ☒

Report (2000 words +/- 100) included in zip ☒

Report length: 2050 words

The code, video, report and marksheet should be zipped and uploaded to the submission link.

I confirm all work done is my own, where third-party sources have been used, I have referenced them appropriately.

Signed: SUMAYA

Notes and Useful Information

This app is entirely client-side with no server required at all – it should run entirely in Google Chrome without any additions.

Example data can be pulled down into your database, which can be extended beyond the data given and updated by the user. No data needs to be written locally but a dummy function can be available which would in-effect update a web database via a server. This does not have to be implemented.

You should deploy and test on Google Chrome browser, using pure JavaScript or compiled TypeScript

Google Chrome will not utilise Cookies locally (file: etc.)

Your code should run offline as well as online (PWA type app, assets locally stored and working preferably using service workers)

Large amounts of data should be stored in the indexedDB database

Web assets should be stored in the cache

You can use cookies or local storage for small bits of information but be aware of the Google Chrome limitation on cookies in the local filing system

You should *fetch* the JSON data stored on the server for doctors (https:/simulacra.uk/CST2572/doctor.json), patients (https:/simulacra.uk/CST2572/people.json) and admin (https:/simulacra.uk/CST2572/admin.json) and medicines (https:/simulacra.uk/CST2572/medicine.json). This data should be processed and placed in your indexeddb database