*Q1 Give an example of a software component you have designed and written from concept to deployment, outlining the steps?*

During my time training at CodeClan, I did a group JavaScript project with 3 other members of the cohort, the outcome of this project is to design a web map app using Vanilla JS with a leaflet map API to pinpoint the Geolocation of known cryptic.

1. Started by using UX **Persona** to identify the target user group

2. Creating **User Story** to understand the user’s behaviours and needs

3. Creating **User Journey** to build a blue point of how the user is interacting with the App

4. Creating a **Wireframe** diagram based on the User Journey

5. Using **RESTful** routes to design frontend and backend

6. Set up **Express Server**

7. Set up **MongoDB** and seeded the Database

8. Set up **PubSub pattern** on the front end between **Model** and **View**

9. Implement Leaflet API on the MapView

This project was completed in a week time, using **Agile**, with **daily stand-up**, **half day sprint**, heavily using **KanBan** and **Trello**, plus **pair programming**, and individual tasks too.

You can check the project: <https://github.com/jo-emalo/js_group_project>

*Q2 Using above example, tell us a significant decision you made to solve a technical challenge given details of the technologies that you chose and why?*

One of the major challenges we had faced was trying to make the sidebar displaying, it was a crucial feature, a lot of the future features have relied on it.

We planned an afternoon sprint to try to learn this additional add-on from leaflet API, after following the official tutorial and learning the basic concept of how the sidebar work, we failed to make it display. At the end of the sprint, I took the responsibility to carry on working on trying to resolve this issue.

After going over the tutorial again by myself, I realised the example code was presented in a chunky block, with a broad definition. So what I decided to do was that **break down the code line by line**, and trying to **understand each line** of the code; the step I took was very simple, I just **console logged everything** one line at the time, by doing that, I realise we called the display sidebar function before the sidebar was fully created. After I put the display function in the right place, the sidebar started to display.

It was a very simple fix, but what it meant to me was that I **gained confidence in working as a team**, and **earned trust from other team members**.

*Q3 using the above example, tell us how you ensure your software was fit for purpose and of high quality, what did you learn and what would you do different time to do a better job?*

Throughout this project we have been focusing solely on designing for the user, we all believe that meeting the **user’s requirement** was the end goal that we were trying to achieve.

To ensure our app was fit for purpose, we asked my landlord’s kids (one boy 8, one girl 5) to be the **user testing group** to give us feedback after using the app, their **feedback** was great, based on their feedback we managed to change one of the features when the map was on the global view, all the pin would show up with the name of the cryptic displayed, and their suggestion was to make the name hidden automatically, only display it when the user hovers the mouse over a particular pin.

At CodeClan I have learned and used **TDD** throughout the course, I do believe bypassing unit tests individually and collectively to ensure that a function or a method is created to do the exactly what it was designed to do, but in this project, we didn’t find a need to use it.

In term of what did I learned from this project, I think my biggest contribution was forcing myself to **pull my ego away** from this project and to contribute to what the team needed me to do. In term of what I could do differently, I think if at the beginning, when the team was formed, I could maybe want to **introduce myself** and my ability to the team, so it would be a better starting point.