III

1

I completed the survey, before doing this task, I ended up using AI tools

2,

19 hours

Both labs: 6,

Learning how to use AWS though the tutorials 7

Actual work on the project: 6

Most of the time was debugging and learning as this is my first time with AWS deployment, and Kotlin and Typescript

3

15%, near the end of the work, I used AI quite a bit as I was starting to get burnt out. However for the majoirty of the process as I wanted to learn I did not use the tools

4

4.1

1. – using Kotlin for debbuging my event feature, built in Gemini for Android APK – Gemini [version unknown] – uknown [more on this later]
2. – media system debugging and feature help – visual studio code embedded chatgpt - unknown > 30 minutes
3. – debugging node & aws for the backend tutorial (helping me find a dumb bug) – Gemini 2.5 pro – 5 minutes

4.2

1. Hoping to debug the kotlin items, my goal was to solve my issues in the shortest time possible. My strategy was to throw the relevant items and review every single line of code so I could actually learn anything. This was very much a moment of desperation for me as I was getting stuck. Using these tools was helpful for a little bit, but for the most part they got stuck in stupid circular logic that did not help. However when they did help it was quick.
2. This was my least focused AI usage of the project, I wanted to fix the bug in the media system, so I reviewed fairly lazly and had the AI look closer on the spots I wanted. Issue: AI's make up stuff. The advantage of an AI,

is that it can create lots of things to test. The disadvantge is that I want the correct solution

3. Debugging an AWS bug for the backend tutorials. I firstly link the backend tutorials here as I ported code over to get my stuff running on AWS. I had a bug with node that I could not sort out. After 2 hours of trying to fix it (and finding other bugs) I gave up and had the AI solve it for me. While this reduced the learning potential they are very good at taking 10k lines of a bug report and pin-pointing the exact errors.