**What I have learned and My experience:**

In my Questioning and Answering application, I have very little idea about design patterns before starting this course, Now I feel I know the working of major design patterns and the usages. In my application, when I started it, I started with the design first based on the features and tried to optimise and structure the features based on patterns and tried to find patterns that are suitable based on required features.

I have used some of the design patterns which I have learned in the course and read from the book. I feel we can develop an application easily, but considering design patterns from creation, structural, behavioural I got to know how we can design more flexible and resilient to change and easy to maintain the application. And I got to know we don't need to reinvent the wheel for common design problems, I feel when I am adding new functionality to my application the changes are very minimal because of design patterns usage I feel it's a kind of standard template we use for the common problems.

**Challenges faced and Things learned from those:**

The major challenges I faced are from choosing the patterns according to the application features I know some patterns when I kept those patterns in code, I used to get errors and null pointer exceptions because sometimes we cannot instantiate an abstract class but there is no concrete there to access the methods inside abstract. And initially starting with the project my idea is to combine all patterns with the features where it worked for a couple of patterns and not worked for other patterns, so, I have planned to do features with pattern separately and integrating to other patterns without doing mix and match.

I tried to integrate patterns like singleton but the use case is not matching to functional requirements, and bridge pattern also. I feel finding the design pattern is challenging according to the requirement and should have a very solid knowledge of all the patterns available.

And also connecting the dots with design patterns is challenging for me, I feel the patterns are overloaded in the application and I didn’t find a way to integrate the new pattern and I lost on the feature's implementation by working around design patterns. And I faced challenges with reusing the code in classes. Because for patterns I have declared abstract classes and having child classes, but with patterns accessing to direct class level methods got challenging because I need to create multiple concrete classes to complete a use case for creating a question and answer for example.

Another challenging problem I faced is after writing some amount of code, I got to know some patterns are not working for the scenarios, and I changed those in the middle which killed some amount of time for features and patterns validation and later I have designed a UML class diagram to visualise the solution with the design pattern.

And I feel some patterns need more solid knowledge to validate for implementations. But with the creation of UML class diagrams, I am able to visualise and understand how it going to work before implementation.