The above structures and layouts show some of the possible ways we can architect our solutions, and also illustrate the differences between layers and tiers. We can have more tiers (n-tier), and can customize our solution with a mix of tiers and layers, according to the project's needs. There is a common misconception among beginner developers that a 3-tier (or n-tier) architecture is the only best model, and many new developers try to blindly follow this model without even giving a second thought to their actual project's needs. As we go from one tier to n-tier, the code complexity increases, and it is better not to go for an n-tier architecture unless the application demands it. For small projects, we can keep things simple and easy.

In the coming chapters we will learn how, why, and which architecture to use (with sample projects), depending on the business needs.

Summary

In this chapter, we learnt the definitions of architecture and design, how they are different from each other and where they fit into our projects. It is very important to understand the different stages of a project life cycle so that we can manage our projects better and mitigate risks early. We also examined the difference between tiers and layers and the different ways we can structure our project using tiers and/or layers. In the coming chapters, we will go deeper into n-tier projects and, with sample applications and code, we will understand the advantages and disadvantages of each option.