**Chapter 2**

In Chapter 2, I learned how to set up Flutter and create a simple "Hello World" application. The process began with installing Flutter SDK, setting up Visual Studio Code or Android Studio, and configuring an Android Emulator for testing. I learned the importance of using flutter doctor to ensure the development environment is properly configured. Creating a new Flutter project with the flutter create command helped me understand the default project structure. I also launched the Android Emulator to test my app and explored the flutter run command to compile and display the app on the emulator. This setup provided the tools necessary to start building with Flutter.

Next, I learned how to create a "Hello World" app and explored the basics of Flutter’s widget-based framework. Editing the main.dart file allowed me to replace the default code with a simple app that displayed the text "Hello World." I learned about the role of the runApp() function in initializing the app and how the MaterialApp widget provides a foundation for Flutter applications. Using the Scaffold widget, I structured the app’s layout, and the Text widget helped display content on the screen. I also learned about the widget tree and its importance in organizing and designing the app's user interface. These concepts introduced me to Flutter's declarative UI approach.

Finally, I learned the significance of Flutter's hot reload feature for efficient development. Hot reload allowed me to make changes to the app’s code and see the results instantly without restarting the app. I tested this feature by modifying the text in the app and observing the updated output in real-time. This feature made experimenting with the app's design and functionality easier and faster. By the end of the chapter, I had successfully set up Flutter, built a basic app, and gained confidence in using hot reload to speed up the development process.