

Flutter in 7 Days

*Build user-friendly apps with
widgets and navigation*

Ipsi Patro



www.bpbonline.com

First Edition 2024

Copyright © BPB Publications, India

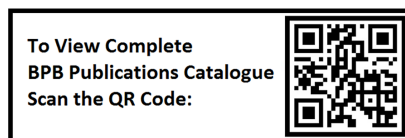
ISBN: 978-93-55516-091

All Rights Reserved. No part of this publication may be reproduced, distributed or transmitted in any form or by any means or stored in a database or retrieval system, without the prior written permission of the publisher with the exception to the program listings which may be entered, stored and executed in a computer system, but they can not be reproduced by the means of publication, photocopy, recording, or by any electronic and mechanical means.

LIMITS OF LIABILITY AND DISCLAIMER OF WARRANTY

The information contained in this book is true to correct and the best of author's and publisher's knowledge. The author has made every effort to ensure the accuracy of these publications, but publisher cannot be held responsible for any loss or damage arising from any information in this book.

All trademarks referred to in the book are acknowledged as properties of their respective owners but BPB Publications cannot guarantee the accuracy of this information.



Dedicated to

*To my beloved husband, Sanat Patro,
and my two sons, Arnav Patro and Vivaan Patro*

About the Author

Ipsi Patro is a software engineer with extensive experience in mobile platforms, excelling in both native and hybrid environments. She graduated with a B.Tech degree and has since worked at renowned companies such as Tata Consultancy Services in India, Vitality, and OvalMoney in the UK, among several others. Currently serving as the Head of Engineering at Twyn in the UK, Ipsi has consistently showcased outstanding leadership and managerial skills, effectively guiding mobile development teams with proficiency.

Motivated by an insatiable curiosity and a dedication to remaining at the forefront of technology, Ipsi immersed herself in mastering Flutter and Dart, swiftly achieving proficiency in both. Her technical expertise spans various programming languages, frameworks, and tools, complemented by a deep understanding of software architecture, agile methodologies, and DevOps practices. These capabilities have empowered her to successfully oversee complex projects from conception through deployment.

Beyond her technical accomplishments, Ipsi is passionate about knowledge-sharing. Her extensive hands-on experience with Flutter projects, coupled with a natural talent for teaching, enables her to articulate complex concepts clearly and accessibly. She firmly believes in the transformative impact of education and is committed to empowering aspiring developers to fulfill their potential.

When not deeply engaged in coding or teaching, Ipsi enjoys pursuits such as listening to music, gardening, cooking, and exploring the outdoors. Actively involved in developer communities, she collaborates with peers to exchange insights and deepen her knowledge.

Ipsi's book, **Flutter in 7 Days**, reflects her deep expertise and passion for Flutter development. She hopes this guide will empower readers to embark on their own Flutter journey and create exceptional mobile applications.

About the Reviewer

SriSindhu Pydimukkala is a seasoned software engineer with a strong foundation in Android development and a burgeoning expertise in Flutter. With seven years of experience in Android development, SriSindhu brings a wealth of knowledge and a fresh perspective to the world of cross-platform mobile app development.

Graduating with a B.Tech degree from Jawaharlal Nehru Technological University, SriSindhu embarked on a dynamic career journey, collaborating with industry-leading companies such as *OvalMoney* and presently serving as a Staff Software Engineer at *BlueYonder*. Throughout her professional tenure, SriSindhu has demonstrated exceptional leadership skills, having led mobile development teams with finesse and efficiency.

SriSindhu's proficiency spans a wide spectrum of Android development, encompassing frameworks, app architecture, memory management, multi-threading, development tools, debugging, problem-solving, Test-Driven Development (TDD), Pair Programming, and deployment strategies. Her commitment to excellence and knack for taking ownership of projects have propelled her to success in delivering high-quality Android applications that meet and exceed client expectations.

Outside of her professional endeavors, SriSindhu enjoys cooking up culinary delights in the kitchen, embarking on adventures to new destinations around the globe, and cherishing quality time with family. Her passion for technology is matched only by her love for exploring new flavors, cultures, and experiences, making her a well-rounded individual with a thirst for knowledge and a zest for life.

Acknowledgement

I would like to express my deepest gratitude to my family and friends for their unwavering support and encouragement throughout the writing of this book. A special thanks to my beloved husband, Sanat Patro, and my two sons, Arnav Patro and Vivaan Patro, whose love and patience have been my greatest source of strength.

I am also profoundly grateful to BPB Publications for their guidance and expertise in bringing this book to fruition. This journey involved numerous revisions and benefited immensely from the participation and collaboration of reviewers, technical experts, and editors.

I would like to acknowledge the valuable contributions of my colleagues and co-workers during my many years in the tech industry. Their insights and feedback have significantly shaped my understanding and approach to mobile app development.

Finally, I extend my heartfelt thanks to all the readers who have shown interest in my book. Your support and enthusiasm have been invaluable in making this book a reality. Thank you for embarking on this journey with me.

Preface

Welcome to *Flutter in 7 Days*! If you are holding this book, you have taken the first step towards mastering one of the most versatile and powerful frameworks for building cross-platform applications. Whether you are a seasoned developer looking to expand your skill set or a newcomer eager to dive into the world of mobile app development, this book is designed to be your comprehensive guide to learning Flutter in just seven days.

In today's fast-paced world, the demand for high-quality mobile applications is ever-increasing. Flutter, with its robust set of features and unparalleled flexibility, has emerged as a frontrunner in the realm of cross-platform development. With Flutter, you can create stunning user interfaces, leverage a single codebase to deploy apps on multiple platforms, and enjoy blazing-fast performance – all without compromising on quality or user experience.

Flutter in 7 Days is structured to provide you with a step-by-step journey through the fundamentals of Flutter development. Each day is carefully crafted to build upon the concepts learned in the previous days, ensuring a smooth and progressive learning experience. From setting up your development environment to building complex UI layouts, handling user input, and integrating with backend services, this book covers everything you need to know to become proficient in Flutter development.

Throughout this book, you will find a balance of theoretical explanations, practical examples, and hands-on exercises to reinforce your learning. Whether you prefer to read through concepts, follow along with code examples, or roll up your sleeves and dive into coding challenges, you'll find something here to suit your learning style.

As you embark on this seven-day journey, remember that learning Flutter – like any new skill – takes time, patience, and practice. Do not be discouraged by challenges or setbacks along the way. Embrace them as opportunities to deepen your understanding and grow as a developer.

By the end of *Flutter in 7 Days*, you will have the knowledge and confidence to tackle real-world Flutter projects with ease. Whether you are building your own apps, contributing to open-source projects, or launching your career as a Flutter developer, this book will serve as your trusted companion on your Flutter journey.

So, without further ado, let us embark on this exciting adventure into the world of Flutter development. Get ready to unleash your creativity, build amazing apps, and take your skills to new heights. The next seven days are going to be both challenging and rewarding – let us make the most of them!

Chapter 1: Day 1– Getting Started with Flutter and Dart

This chapter introduces Flutter and Dart, guiding you through setting up your development environment and creating your first Flutter project. You will gain an understanding of the Flutter project structure and write your first Dart code.

Chapter 2: Day 2– Basic Programming Concepts

In this chapter, you will learn about variables and data types in Dart, control flow mechanisms such as loops and conditionals, and the fundamentals of functions and methods. Additionally, you will explore object-oriented programming basics in Dart and engage in hands-on exercises to reinforce these concepts.

Chapter 3: Day 3– First Flutter App

This chapter delves into the Flutter widget tree, teaching you how to create layouts with widgets and handle user input through gestures and interactions. You will learn to navigate between screens and build a simple app from scratch.

Chapter 4: Day 4– Flutter Widgets

Here, you will get an overview of commonly used Flutter widgets, including container widgets for layout and styling, and text and image widgets for displaying content. The chapter also covers ListView and GridView for creating scrolling lists and grids, as well as exploring advanced widgets and customizing their behavior.

Chapter 5: Day 5– Prep Up with Advanced Flutter

This chapter focuses on state management in Flutter, understanding the widget lifecycle, and using packages and plugins to extend Flutter’s functionality. Additionally, you will explore Flutter’s animation capabilities and learn techniques for optimizing performance and debugging.

Chapter 6: Day 6– Fetching Data Internet

In this chapter, you will learn how to make HTTP requests in Flutter and handle asynchronous operations using Futures and Streams. You will also learn to parse JSON data, display remote data in your Flutter app, and implement error handling and loading indicators.

Chapter 7: Day 7– Firebase Integration to Flutter App

This chapter introduces Firebase and its services, guiding you through setting up Firebase for your Flutter project. You will learn to integrate Firebase Authentication, store and retrieve data with Cloud Firestore, and implement real-time updates with Firebase Cloud Messaging.

Chapter 8: Miscellaneous

The final chapter offers tips and best practices for Flutter development, guidance on deploying your Flutter app to different platforms, and resources for further learning and exploration. It also addresses troubleshooting common issues and exploring advanced topics and next steps in your Flutter journey.

Code Bundle and Coloured Images

Please follow the link to download the
Code Bundle and the *Coloured Images* of the book:

<https://rebrand.ly/077b4d>

The code bundle for the book is also hosted on GitHub at

<https://github.com/bpbpublications/Flutter-in-7-Days>.

In case there's an update to the code, it will be updated on the existing GitHub repository.

We have code bundles from our rich catalogue of books and videos available at **<https://github.com/bpbpublications>**. Check them out!

Errata

We take immense pride in our work at BPB Publications and follow best practices to ensure the accuracy of our content to provide with an indulging reading experience to our subscribers. Our readers are our mirrors, and we use their inputs to reflect and improve upon human errors, if any, that may have occurred during the publishing processes involved. To let us maintain the quality and help us reach out to any readers who might be having difficulties due to any unforeseen errors, please write to us at :

errata@bpbonline.com

Your support, suggestions and feedbacks are highly appreciated by the BPB Publications' Family.

Did you know that BPB offers eBook versions of every book published, with PDF and ePub files available? You can upgrade to the eBook version at www.bpbonline.com and as a print book customer, you are entitled to a discount on the eBook copy. Get in touch with us at :

business@bpbonline.com for more details.

At **www.bpbonline.com**, you can also read a collection of free technical articles, sign up for a range of free newsletters, and receive exclusive discounts and offers on BPB books and eBooks.

Piracy

If you come across any illegal copies of our works in any form on the internet, we would be grateful if you would provide us with the location address or website name. Please contact us at business@bpbonline.com with a link to the material.

If you are interested in becoming an author

If there is a topic that you have expertise in, and you are interested in either writing or contributing to a book, please visit www.bpbonline.com. We have worked with thousands of developers and tech professionals, just like you, to help them share their insights with the global tech community. You can make a general application, apply for a specific hot topic that we are recruiting an author for, or submit your own idea.

Reviews

Please leave a review. Once you have read and used this book, why not leave a review on the site that you purchased it from? Potential readers can then see and use your unbiased opinion to make purchase decisions. We at BPB can understand what you think about our products, and our authors can see your feedback on their book. Thank you!

For more information about BPB, please visit www.bpbonline.com.

Join our book's Discord space

Join the book's Discord Workspace for Latest updates, Offers, Tech happenings around the world, New Release and Sessions with the Authors:

<https://discord.bpbonline.com>



Table of Contents

1. Getting Started with Flutter and Dart	1
Introduction.....	1
Structure.....	1
Objectives.....	2
Introduction to flutter and dart	2
Understanding Flutter	5
Defining Dart	5
<i>Flutter Architecture</i>	5
Installation in windows	6
<i>Installing Flutter SDK</i>	6
<i>Installing Android Studio</i>	8
Installation in Mac.....	11
<i>Installing Flutter SDK</i>	12
<i>Installing Android studio</i>	13
<i>iOS Setup</i>	14
<i>Updating some configurations for Android Studio</i>	15
IDEs for Flutter	17
<i>Features of IDE</i>	17
Compilation logic of Flutter code	20
Dartpad	20
Function	20
Dart data types.....	21
Dart variables	21
Dart classes and objects	22
Dart List	23
Dart map	24
Null safety in Dart.....	25
<i>Non-nullable types</i>	25
<i>Nullable types</i>	25
String Interpolation.....	26

Further reading	26
Conclusion.....	26
Questions	27
<i>Answers</i>	27
Multiple choice questions.....	28
<i>Answers</i>	28
2. Basic Programming Concepts.....	29
Introduction.....	29
Structure.....	29
Objectives.....	29
Object oriented programming	30
<i>Class</i>	30
<i>Object</i>	31
<i>Inheritance</i>	33
<i>Polymorphism</i>	34
<i>Advantage of polymorphism in Dart</i>	35
<i>Interface</i>	36
<i>Abstract class</i>	38
<i>Data encapsulation</i>	40
<i>Dart mixins</i>	42
<i>Enum</i>	44
<i>Type inference</i>	47
<i>Generics</i>	48
<i>Anonymous functions</i>	50
Loops	51
For loop.....	52
for in	52
While loop.....	53
Do while loop.....	54
VoidCallBack.....	55
Function(n).....	57
Arrow functions.....	60
Ternary operator	61

Introduction to Git.....	62
<i>Difference between Git and Git hosting tool</i>	63
Async await.....	64
<i>Future</i>	66
<i>Stream</i>	66
<i>Completed</i>	66
<i>Uncompleted</i>	66
Conclusion.....	68
3. First Flutter App.....	69
Introduction.....	69
Structure.....	69
Objectives.....	70
Build your first Flutter project.....	70
<i>Folder structure</i>	73
<i>pubspec.yaml</i>	73
<i>pubspec.lock</i>	75
Code formatting tips	75
<i>Closing labels</i>	78
<i>Debug banner</i>	79
Sample app.....	80
Widget.....	81
<i>Widget tree</i>	82
<i>MaterialApp</i>	83
<i>Scaffold</i>	83
Run the app in a simulator / emulator	84
Run the app in a physical device	84
<i>Running on Android device</i>	84
<i>Deploy app to iOS devices</i>	92
App icon.....	95
Hot Reload.....	98
Conclusion.....	100
Multiple Choice Questions.....	101
<i>Answers</i>	101

4. Flutter Widgets	103
Introduction.....	103
Structure.....	103
Layout widgets	104
<i>Single-child layout widgets</i>	104
<i>Container widget</i>	104
<i>Safe area</i>	107
<i>Margin and padding</i>	108
<i>Padding widget</i>	111
<i>Centre widget</i>	112
<i>Align widget</i>	114
<i>Multi-child layout widgets</i>	117
<i>Column</i>	117
<i>Row</i>	120
<i>Stack</i>	122
<i>List View</i>	125
<i>GridView</i>	127
<i>Tab Bar</i>	129
Sliver Widget.....	130
Asset image and network image widget.....	132
<i>Asset image</i>	132
<i>Network image</i>	133
Card Widget	134
Inkwell widget	135
Widget styling	135
Stateless and stateful widgets.....	136
Material and Cupertino widgets	142
<i>Material widgets</i>	142
<i>Cupertino widgets</i>	143
Divider widget.....	144
Slider widget	145
DropDownButton widget	146
Bottom sheet Widget.....	148

ListTile Widget	150
Animations in Flutter	152
Conclusion	158
Multiple choice questions	158
<i>Answer</i>	160
Exercises	160
5. Prep Up with Advanced Flutter	165
Introduction	165
Structure	165
Flutter themes	166
.of and .copyWith operator:	174
.of operator	174
.copyWith operator	176
Custom widgets	178
Flutter packages	183
Gesture detection	186
Pass function as parameter	190
Routing and navigation	193
Mixin	203
Selecting platform specific widget	205
Data persistence in Flutter apps	207
Conclusion	209
Multiple choice questions	210
<i>Answers</i>	211
6. Fetch Data from Internet	213
Introduction	213
Structure	213
Objectives	214
Network call with http library in Flutter	214
Futures in Flutter	217
<i>Future</i>	217
<i>FutureBuilder</i>	217

<i>Async/Await</i>	218
Parsing JSON.....	219
Spinner widget.....	226
Passing data backwards through navigator.....	227
<i>Sending data from the destination screen</i>	227
<i>Receiving data in the previous screen:</i>	227
Location service.....	228
Sample app with networking call.....	230
Conclusion.....	240
Multiple choice questions.....	240
<i>Answers</i>	241
7. Firebase Integration to Flutter App.....	243
Introduction.....	243
Structure.....	243
Objectives.....	244
Introduction to Firebase.....	244
<i>Defining Firebase</i>	244
<i>Key Firebase services</i>	244
<i>Advantages of selecting Firebase for Flutter</i>	245
Setting up Firebase project.....	245
<i>Adding Flutter app to Firebase project</i>	249
Firebase authentication.....	257
Cloud Firestore integration.....	269
CRUD operations with Cloud Firestore in Flutter.....	271
Dart Streams.....	279
StreamBuilder.....	281
Async Snapshot.....	284
QuerySnapshot.....	285
Conclusion.....	288
Multiple Choice Questions.....	289
<i>Answers</i>	290

8. Miscellaneous	291
Introduction.....	291
Structure.....	291
Introduction to the Provider package.....	292
Riverpod	294
Flutter versions	296
Design patterns in Flutter.....	297
<i>Model-View-ViewModel</i>	298
<i>Model-View-Presenter</i>	299
<i>Model-View-Controller</i>	299
Business Logic Component design patterns	300
Unit testing	303
<i>Tips for writing unit tests</i>	306
Debugging	306
Publishing app in an app store.....	307
Publish your Flutter app on the Play Store.....	309
Conclusion.....	311
Index	313-317

CHAPTER 1

Getting Started with Flutter and Dart

Introduction

Hello readers welcome onboard to the journey of becoming a Flutter developer. In this chapter, first we will go through the installation steps to bring your machine ready for Flutter development and then we will go through some basic concepts of Flutter and Dart.

Structure

The chapter covers the following topics:

- Introduction to Flutter and Dart
- Understanding Flutter
- Defining Dart
- Installation in Windows
- Installation in Mac
- IDEs for Flutter
- Compilation logic of flutter
- Dartpad
- Function
- Dart data types

- Dart Variables
- Dart Classes and Objects
- Dart List
- Dart Map
- Null Safety in Dart
- String Interpolation

Objectives

By the end of this chapter, your machine must be ready for Flutter development, and you will get a good understanding of which IDE to pick and some important features provided by the IDE to make your development easier and faster. Along with that, some important and fundamental Flutter and Dart concepts will also be discussed.

Introduction to flutter and dart

In this book, we will learn Flutter from scratch to build fast, beautiful, native-quality iOS and Android Apps using just one code base and one programming language called **Dart**. **Flutter** is a tool developed by Google that can be used to build cross-platform iOS, Android, and web apps. It comes with loads of pre-built widgets, which makes it easier to layout any app. Flutter widgets are built using a modern framework inspired by Web Design.

Flutter just asks for a blank window and draws the widgets in it. Everything in a Flutter app is a widget, so we build a widget upon a widget. You can create your own widget or customize the pre-built widgets as per your requirements. One good thing about Flutter is when you are working with it, you get access to the original source code because it is open-source. If you want to know how an App bar is implemented, just click on the widget, and you can view it. If you are from an iOS development background, you must have noticed that the same thing is not possible in iOS. In iOS, many components are proprietary, so you will not be able to access them.

The term Flutter refers to two major things:

- The SDK contains many tools to compile the dart code to native machine code, which can run on iOS and Android platforms.
- It is a widget library that helps build beautiful apps.

In addition to Flutter, Dart is a fundamental aspect of app development with Flutter. Dart is the programming language utilized for writing code in Flutter. It was developed by Google and is designed for building fast, scalable, and efficient applications. Dart is a statically typed language with a syntax that is familiar to those who have experience with languages like Java or JavaScript.

Here are some key features of Dart:

- **Object-oriented:** Dart is an object-oriented language, which means it supports concepts like classes, objects, inheritance, and polymorphism. This makes it easy to organize code and build complex applications.
- **Strongly typed:** Dart is a statically typed language, meaning that variables are explicitly declared with their data types. This helps catch errors at compile-time and improves code robustness.
- **Asynchronous programming:** Dart provides built-in support for asynchronous programming using features like `async` and `await`. This allows developers to write code that can handle multiple tasks concurrently, such as making network requests or performing file I/O operations, without blocking the main execution thread.
- **Garbage collection:** Dart includes automatic memory management through garbage collection. Developers do not need to manually allocate and deallocate memory, making memory management less error-prone.
- **Package management:** Dart has a robust package management system called Pub. Developers can use Pub to easily manage dependencies and share their own packages with the community.
- **Cross-platform development:** While Dart is primarily associated with Flutter for mobile app development, it can also be used for server-side and web development. This versatility allows developers to use Dart for a wide range of projects beyond just mobile apps.

Understanding Dart is essential for effectively utilizing Flutter's capabilities to build powerful and feature-rich mobile applications. By mastering Dart, developers can leverage their strengths to write clean, efficient code that powers Flutter apps across multiple platforms.

Let us delve into a comprehensive comparison between Flutter and other popular frameworks to understand their respective strengths, weaknesses, and unique features in the realm of cross-platform mobile app development.

- **Flutter vs. React Native**
 - React Native, developed by *Facebook*, allows developers to build cross-platform mobile apps using JavaScript and React.
 - In Flutter, everything is a widget, which provides a more consistent and streamlined development experience compared to React Native's mix of native components and JavaScript.
 - Flutter uses a compiled programming language (Dart), which can result in better performance and more predictable behavior compared to React Native's JavaScript runtime.

- Hot Reload in Flutter offers a faster development cycle compared to React Native's Hot Reload, as it updates the entire UI tree rather than just the JavaScript code.
- **Flutter vs. Xamarin**
 - Xamarin, owned by *Microsoft*, enables developers to build cross-platform mobile apps using C# and .NET.
 - Flutter's UI is rendered using Skia, a high-performance 2D graphics library, which can result in smoother animations and transitions compared to Xamarin's native UI approach.
 - Flutter offers a more consistent development experience across platforms since it uses a single codebase, while Xamarin requires separate UI code for each platform.
 - Dart's **Ahead-of-Time (AOT)** compilation in Flutter can lead to faster startup times and reduced runtime overhead compared to Xamarin's **Just-in-Time (JIT)** compilation.
- **Flutter vs. Native Development (iOS/Android)**
 - Native development involves using platform-specific languages (Swift/Objective-C for iOS, Java/Kotlin for Android) and APIs to build apps.
 - Flutter offers faster development cycles and a more productive workflow compared to native development, thanks to features like Hot Reload and a single codebase for multiple platforms.
 - While native development provides access to platform-specific APIs and features, Flutter's extensive widget library and platform channels allow developers to achieve native-like behavior and performance.
 - Flutter's UI is rendered using Skia, which may result in slightly different visual rendering compared to native components, though Flutter's customizable widgets can often achieve the desired look and feel.

In summary, Flutter's use of a single codebase, fast development cycles, high-performance rendering engine, and customizable widgets make it a compelling choice for cross-platform mobile app development, offering unique advantages over other frameworks like React Native, Xamarin, and native development. Additional advantages of Flutter, such as Hot Reload and strong community support, can provide further insights into its appeal. Here is an expanded comparison with these advantages:

- **Hot Reload for Rapid Development:**
 - Flutter's Hot Reload feature allows developers to instantly see changes made to the code reflected in the app's UI without restarting the app or losing its current state.

- This iterative development process significantly reduces development time by enabling developers to experiment, iterate, and refine UI elements and features rapidly.
- In contrast, other frameworks like React Native and Xamarin offer similar features, but Flutter's Hot Reload is often praised for its speed and reliability, making the development experience smoother and more efficient.
- **Strong community support:**
 - Flutter benefits from a large and active community of developers, enthusiasts, and contributors who provide extensive support, resources, and collaboration opportunities.
 - The Flutter community actively shares knowledge, best practices, libraries, packages, and plugins through platforms like GitHub, Stack Overflow, Medium, and various forums and meetups.
 - This vibrant community ecosystem fosters innovation, accelerates learning curves, and addresses challenges faced by developers, making it easier to find solutions and resources for building Flutter apps.
 - Additionally, Google's continued investment in Flutter and its commitment to open-source development further bolster community engagement and contribute to the framework's growth and evolution.

Understanding Flutter

Google has introduced Flutter for native mobile app development on Android, iOS, and Windows. Flutter is a mobile app SDK, complete with framework, widgets, and tools, that gives developers a way to build and deploy mobile apps written in Dart. So, before moving ahead with learning Flutter, we will cover Dart basics later in this chapter. If you have prior experience of software development with any language, you must be familiar with these basics, but if you do not have any, it will be helpful for you in this journey.

Defining Dart

Dart is a programming language designed for client development, such as for the web and mobile apps. It is developed by Google and can also be used to build server and desktop applications. It is an object-oriented, class-based, garbage-collected language with C-style syntax.

Flutter Architecture

With UI as code, there is no visual editor in Flutter like native iOS and Android development. Therefore, no drag and drop of widgets; instead, write the code for adding and updating