Rapid SaaS Application Development Using Salesforce

Build scalable SaaS applications using the Salesforce platform

Tameem Bahri

www.bpbonline.com

Every effort has been made in the preparation of this book to ensure the accuracy of the information presented. However, the information contained in this book is sold without warranty, either express or implied. Neither the author, nor BPB Online or its dealers and distributors, will be held liable for any damages caused or alleged to have been caused directly or indirectly by this book.

BPB Online has endeavored to provide trademark information about all of the companies and products mentioned in this book by the appropriate use of capitals. However, BPB Online cannot guarantee the accuracy of this information.

First published: 2024

Published by BPB Online
WeWork
119 Marylebone Road
London NW1 5PU

UK | UAE | INDIA | SINGAPORE

ISBN 978-93-55519-009

www.bpbonline.com

Dedicated to

My wife, ***Lana****, who is my best friend and my greatest supporter.*

My son, ***Ameer****, the light of my life.*

My daughter, ***Lea****, who fills my heart with joy each and every day.*

Every teacher who ignites a spark in their students.

All those who pick up this book, thank you for
giving my words a chance.

About the Author

Tameem Bahri is the Technology Leader for the CRM and CIAM platforms at A.P. Moller - Maersk and was formerly the European Chief Technology Officer for the Salesforce COE at Capgemini. He is an experienced technology specialist with a demonstrated history of working in the information technology and services industry with over 20 years of experience across Business Transformation, Digital Services, Innovation, Process design and redesign, Enterprise System Security, **Identity and Access Management** (**IAM**) Strategy and Enterprise Solution Architecture.

Tameem Bahri is a Salesforce **Certified Technical Architect** (**CTA**) and the author of other books concerning Salesforce architecture and solutions. Tameem is a public speaker who participated in multiple events, such as DreamForce, the Data Innovation Forum, and the Tunisia Digital Summit.

About the Reviewer

Subramani Kumarasamy is a Certified Salesforce Application Architect specializing in automating critical business processes within Salesforce. With expertise in end-to-end solutions, he has successfully managed diverse projects of varying complexities. With a passion for delivering technical excellence, he consistently goes above and beyond to provide automated solutions that drive business success. His dedication to building technical debt-free systems, expertise in license management, and mastery of the Salesforce core platform and integration make him a sought-after professional in the Salesforce ecosystem.

Subramani's passion for teaching led him to train numerous professionals globally, both freshers and experienced, in Salesforce. His ability to simplify complex concepts and provide hands-on training has earned him recognition in the Salesforce community.

Beyond his professional pursuits, Subramani enjoys indulging in his favorite television shows, such as Game of Thrones and Suits, during his leisure time. However, his true joy lies in spending quality moments with his family and friends, as he understands the importance of work-life balance and cherishes the relationships that matter most to him.

Acknowledgement

I want to express my deepest gratitude to my family for their unwavering support and encouragement throughout this book's writing.

I am also grateful to BPB Publications for their guidance and expertise in bringing this book to fruition. It was a long journey of revising this book, with valuable participation and collaboration of reviewers, technical experts, and editors.

I would also like to acknowledge the valuable contributions of my colleagues and co-workers during many years working in the tech industry, who have taught me so much and provided valuable feedback on my work.

Finally, I would like to thank all the readers who have taken an interest in my book and for their support in making it a reality. Your encouragement has been invaluable.

Preface

During the twenty-plus years of my career, I had the pleasure of working in the software development, consulting, and service industries. During my time in the software development industry, I had the pleasure of using different technologies to create and fine-tune different types of cutting-edge software, from desktop to web applications, from device drivers to mass IoT-powered distributed solutions, and from simple websites to SaaS applications using the Salesforce platform. I had excellent exposure to some of the finest professionals in software development and architecture, who taught me a lot.

In the past seven years or so, I had the pleasure of coaching multiple dozens of senior Salesforce architects. Many have developed into recognized **Salesforce Certified Technical Architects (CTAs)**. During that journey, I noticed differences between the activities followed by the Salesforce developers and architects and those adopted by their classic software development counterparts. The differences are mainly derived from the nature of the Salesforce platform and its incredible flexibility. Most of the missing concepts and activities are actually required to pass the CTA review board exam (one of the hardest and most thorough exams in the technology industry nowadays). The candidates I coached had to learn new concepts and generate specific artifacts to help them explain a solution end-to-end. Sometimes, the candidates had to unlearn some habits they had used for several years, which further prolonged the time required to prepare for the exam.

I also noticed that companies that fully embrace and adopt the Salesforce platform and extend it to cover different business needs often struggle with technical debts that slow the development pace of future requirements. Slowing down the development pace of Salesforce applications takes away one of the platform's key benefits: time-to-market. Addressing these technical debts is costly and time-consuming; therefore, avoiding them from the beginning would add immense value to the company.

Helping Salesforce professionals rapidly develop scalable, performant, and secure Salesforce applications while avoiding technical debt would benefit both the individuals and the companies they work for. This is precisely what this book aims at. This book teaches some essential Salesforce application development habits, activities, best practices, and concepts using a practical, example-led approach.

The Salesforce platform is an extremely powerful and flexible tool. It enabled millions of users to customize and build enterprise-scale applications used daily. The platform's flexibility means it is up to the implementer to use it right and make a scalable, expandable, performant, and secure application, or use it wrong and end up with a suboptimal application that is not aligned with today's agile and fast-paced businesses. Rapid SaaS Application Development using Salesforce shows how to design, document, build, validate, and release an application using a rapid prototyping approach without sacrificing the solution's quality or expandability.

This book is primarily meant for the following readers:

- **CRM enthusiasts** who are looking to learn more about the Salesforce platform and its rapid application development capabilities. For those readers, I recommend chapters 1, 3, 4, 5, and 10.
- **Salesforce citizen developers** who are looking to learn more about low-code/no-code development concepts while bridging the gap with full-code developers. For those readers, I recommend chapters 1 to 10.
- **Salesforce business analysts** who are keen to learn about techniques for excavating uncommunicated requirements and communicate them in a way that avoids misunderstanding by the developers and testers. For those readers, I recommend chapters 1, 3, 4, 5, 9, and 10.
- **Junior Salesforce professionals** who are open to learning some of the best practices in developing secure, scalable, and maintainable Salesforce applications rapidly. For those readers, I recommend chapters 1 to 10.
- **Senior Salesforce professionals** who are looking to refresh some of their past knowledge. For those readers, I recommend chapters 1, 3, 4, 5, 6, 9, and 10.

This book will teach you the structured rapid application development lifecycle for the Salesforce platform, which is derived from the solid software development lifecycle. This book will use real-life business scenarios to teach you foundational information and skills that you can use to develop enterprise-ready, AI-powered applications. I sincerely hope you enjoy this book and find it useful.Parte superior do formulárioParte inferior do formulário Tameem Bahri

Chapter 1: Introduction to the Salesforce Platform - This chapter provides general information about the Salesforce Platform and Salesforce SaaS Applications. The chapter explains the rationale behind creating SaaS applications on top of the Salesforce Platform. The chapter then explains the underlying infrastructure of the Salesforce Platform, which is essential to developing scalable enterprise-grade applications. The reader is then introduced to the Salesforce AppExchange, where applications can be listed in the Salesforce marketplace. The chapter will also briefly introduce the key Salesforce application building blocks and compare the different coding and configuration options supported by the platform. Finally, the reader is introduced to the concept of developing an application with off-platform capabilities offered by other platforms, including Salesforce Heroku.

Chapter 2: Deep Dive into Key Building Blocks and Tools - This chapter dives deeper into the Salesforce application building blocks and explains each in detail using simple examples. The reader will be introduced to Users, Licenses, Objects (standard and custom), Fields (standard and custom), Page Layouts, Sharing and Security models, App Builder, Report Builder, Flow Builder, and more.

Chapter 3: Develop a Sample Salesforce Application: PbP Phonebook - This chapter practically teaches how to build a simple Salesforce Application using point-and-click development tools. The reader will learn about analyzing and understanding requirements, defining process flows, creating prototypes, and building, rectifying, and deploying an application. The reader will then learn about the gaps and mistakes that could occur without a well-structured application development lifecycle.

Chapter 4: Learn the Salesforce Application Development Lifecycle - This chapter introduces the reader to the Salesforce application development lifecycle and compares it to its parent, the standard application development lifecycle, pointing out their differences and the rationale behind them. The reader is then introduced to a set of practices to follow while developing any Salesforce application.

Chapter 5: Understand the Supporting Tools and Artifacts - This chapter introduces a set of artifacts and tools that a Salesforce Application Developer can use to rapidly develop secure, scalable, extendable, flexible, and performant applications while avoiding creating technical debt. The chapter explains the importance of these artifacts and illustrates how they can be created using easy-to-follow examples.

Chapter 6: Create a Sample Application: Define and Refine the Requirements - This chapter practically puts all that the reader learned into action by introducing a real-world business challenge that the reader would solve using a Salesforce SaaS Application. The reader will first be introduced to the business challenge, then proceed to analyze the requirements and identify gaps. The reader will then create testable user stories that will form the basic requirements that will be solved in the two following chapters. The reader will also create high-level process flows and capability maps as part of the application requirement documentation.

Chapter 7: Create a Sample Application: Solve and Build the Application - Part 1 - The reader will start solving the business problem introduced in *Chapter 6, Create a Sample Application: Define and Refine the Requirements*, using the Rapid Salesforce Application Development Lifecycle tools and principles. The reader will solve the identified user stories one by one, creating and updating the relevant artifacts at each stage. The reader will be introduced to the process of identifying multiple solutions for a particular problem and selecting the most appropriate one, considering all the pros and cons.

Chapter 8: Create a Sample Application: Solve and Build the Application - Part 2 - The reader will continue solving the business problem introduced in *Chapter 6, Create a Sample Application: Define and Refine the Requirements*, and continue creating a solution based on the partial solution introduced in *Chapter 7, Create a Sample Application: Solve and Build the Application - Part 1*. The reader will continue using the Rapid Salesforce Application Development Lifecycle tools and principles. The reader will solve the identified user stories one by one, creating and updating the relevant artifacts at each stage. The reader will continue learning about the process of identifying multiple solutions for a particular problem and selecting the most appropriate one, considering all the pros and cons. The reader will also briefly learn about some scalability considerations for the Salesforce Platform.

Chapter 9: Create a Sample Application: Test and Deploy - The reader will deploy and test the application that was developed in *Chapter 7, Create a Sample Application: Solve and Build the Application - Part 1* and *Chapter 8, Create a Sample Application: Solve and Build the Application - Part 2*. The reader will learn about the different deployment tools and techniques and will be introduced to concepts such as CI (Continuous integration) and CD (Continuous Development). The reader will learn about the importance of testing the application and the test case format, types, and management. The reader will learn about tools for each test type and dive deeper into one of the major testing tools, the Apex test classes.

Chapter 10: Tips and Tricks and the Way Forward - This chapter introduces the reader to more advanced topics and will explain the value of release notes. The reader will learn how to create useful and impactful release notes and how to efficiently publish and share them with the right audience. The reader will also learn how to enrich an application backlog using ideas and feedback gathered from your customers directly. Finally, the reader will be introduced to further learning materials and resources to continue learning from.

Code Bundle and Coloured Images

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

https://rebrand.ly/ca55e2

The code bundle for the book is also hosted on GitHub at
https://github.com/bpbpublications/Rapid-SaaS-Application-Development-Using-Salesforce
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.

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

CHAPTER 1

Introduction to the Salesforce Platform

Introduction

This book chapter will introduce you to the **Salesforce platform** (formerly known as **Force.com**) and the **Salesforce App Cloud**. Some still mis-interpret **Salesforce.com** as a **Customer Relationship Management (CRM)** system offered as a **multi-tenancy Software as a Service (SaaS)**. The massive success of Salesforce in the CRM industry might influence such misdefinition. In this chapter, you will learn about another powerful - yet less known - capability of the Salesforce platform as a **Rapid Application Development (RAD)** environment. But first, take a look at the following definitions.

In 1995, the term Customer Relationship Management was coined, then became popular from 1997 to 2000 thanks to the companies such as **IBM** and **Siebel**. At that time, CRM systems included tools that helped to standardize and manage business processes across sales and service. This term extended eventually to include marketing and digital commerce.

Multi-tenancy is a term used to describe a way to run software applications where a single instance of the software (in the Salesforce world, this is called an organization, or for short, org) runs on an infrastructure (such as a server) that is configured to serve multiple users and multiple instances of the software. In a multi-tenancy setup, the host ensures that the data of all tenants are not mixed up (Salesforce does that using a unique identifier for each org called the Org ID). However, tenants might share the same resources the infrastructure offers (such as memory and computing power).

Software as a service (SaaS) is a software delivery and licensing model (also known as on-demand software) where a vendor offers ready-to-use software via a subscription-based licensing model. The SaaS software can usually be configured to meet their end user's needs. The SaaS model simplifies the infrastructure challenges such as hosting, scaling, and maintaining software, as the vendor would be offering such services as part of the subscription fees. SaaS software is typically accessed via a web browser (also known as a thin client).

RAD describes a software development approach that promotes agile, fast-paced, and prototype-driven software engineering compared to traditional waterfall development processes, which are inherited from other conventional engineering models that are not necessarily fit for software.

Structure

This chapter covers the following topics:

- What is a Salesforce application
- Why develop a Salesforce application
- The Salesforce platform infrastructure
- Getting to know the Salesforce AppExchange
- Key building blocks and tools
- No-code vs. low-code vs. full-code
- Salesforce Objects, more than a database
- On-platform and off-platform

Objectives

By the end of this chapter, you will understand what a Salesforce application is and why the Salesforce platform is a very compelling RAD environment. You will better understand the underlying infrastructure of the Salesforce platform and learn about the application marketplace known as **AppExchange**. You will learn about the key building blocks available to developers to develop applications on the Salesforce platform rapidly. You will get terms such as no-code, low-code, and full-code demystified. You will then learn more about the underlying database of the **Salesforce platform**, which used to be known as database.com. You will also learn what type of applications and features are fit to be developed on the Salesforce platform and what type is better-fit off-platform.

What is a Salesforce application

Salesforce applications is an application program stored in the cloud on top of the Salesforce platform and delivered via a browser interface. It is a form of web application delivered as a SaaS using the pre-built capabilities of the Salesforce platform. Salesforce applications are built using many development tools ranging from point-and-click tools to fully customizable code. Point-and-click development can also be referred to as *declarative development* or *declarative configuration*.

The following diagram compares a simplified web application architecture to a simplified Salesforce application architecture:

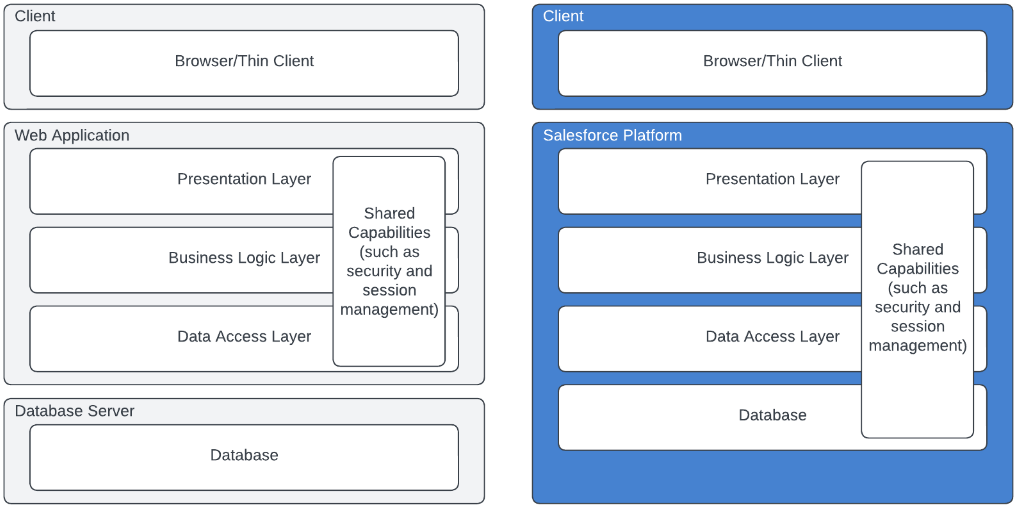

***Figure 1.1**: Comparing web application architecture to Salesforce application architecture*

You might have noticed that the application in both cases is separated into multiple layers. This approach is called **Separation of Concern** (**SOC**). In most cases, you will not need to bother yourself with creating SOC layers while developing Salesforce applications, mainly while using declarative development. Declarative development is faster, simpler, and usually more reliable than custom development. Moreover, it has a built-in SOC structure to some extent. You will learn about the declarative and non-declarative (coding) tools available in the Salesforce platform later in this chapter under the *Key building blocks and tools* section.

The Salesforce platform is part of a bigger organizational structure called the **Salesforce App Cloud**. Here are some of the main components of the Salesforce App Cloud:

- **Salesforce platform**: The heart and soul of the App Cloud and the underlying foundation for several other Salesforce clouds such as Service Cloud, Sales Cloud, and Experience Cloud. The Salesforce platform was formerly known as Force.com. The platform offers capabilities that enable administrators and developers to build secure, performant, and scalable web applications that are delivered using a SaaS model. The Salesforce platform is the primary technology covered in this book. You can think of the Salesforce platform as the classic LEGO blocks that you can use to build any LEGO model. In this analogy, products such as Sales Cloud and Service Cloud would be something like a pre-assembled LEGO Star Wars Millennium Falcon, where the classic building blocks are used to create a fully functional model that you can tweak or extend further. Salesforce also offers out-of-the-box **Artificial intelligence (AI)** products natively integrated with the Salesforce platform. These products are all branded under the **Salesforce Einstein** umbrella.

- **AppExchange**: The **AppExchange.com** website is a marketplace for the Salesforce platform's applications. AppExchange allows developers to publish their work to the marketplace, where it can be tried, installed, used, and reviewed by Salesforce customers. The AppExchange contains thousands of applications. It is a popular destination for Salesforce customers, enabling them to extend their Salesforce implementations with additional pre-built capabilities easily. The applications listed on the AppExchange can range in complexity from simple, single-purpose applications to very complex solutions such as fully-fledged ERPs. Publishing applications to the AppExchange is out of the scope of this book.

- **Heroku**: **Heroku.com** is a cloud platform that allows developers to build, host, run, and monitor their applications in a quick and simplified way. It supports several development languages and database technologies. Heroku is considered a **Platform as a Service (PaaS)**. The clear difference between PaaS and SaaS is that the former offers a set of tools that helps developers build, host, run, and monitor their applications. In contrast, the latter provides a ready-to-use pre-built application that is usually configurable/customizable. The Salesforce platform is much more customizable than a regular SaaS but more restricted and controlled than a classic PaaS. Configuring Heroku and developing applications on it is out of the scope of this book.

- **Salesforce Shield**: Offers services to protect and monitor the customer's data stored on the **Salesforce platform**. Salesforce Shield offers functionalities such as data encryption at rest which is a must for some businesses. Configuring and using Salesforce Shield is out of the scope of this book.

- **Salesforce DX**: This is a set of developer tools that enables building and managing **Salesforce platform** applications' development lifecycle in a controlled and efficient way. Salesforce DX allows developers to easily use **Source control management** (**SCM**) solutions such as Git (also known as version control solutions) to control code versions, collaboration, conflict resolution, auditing, code recovery, and release management. Configuring and using Salesforce DX is out of the scope of this book. All examples in this book will use simpler mechanisms to allow you to focus on the other essential aspects of the Salesforce platform application's lifecycle.

- **Salesforce identity**: Provides a set of trusted services to allow enterprises to manage their employees' and customers' identities. Configuring and using Salesforce identity is out of the scope of this book.
- **Salesforce Trailhead**: the **Trailhead.Salesforce.com** website is one of the leading online learning platforms available today. It contains a plethora of tutorials, exercises, and knowledge articles that teach a broad spectrum of topics to different audiences, such as administrators, developers, business analysts, UX designers, architects, and business owners from all levels of expertise. Salesforce Trailhead is a very handy tool for any Salesforce professional, and you are strongly encouraged to familiarize yourself with it. The Trailhead website will be referred to several times throughout this book.

Next, you will learn some benefits of developing apps on the Salesforce platform.

Why develop a Salesforce application

There are certain benefits of developing applications on the Salesforce platform (also known as Salesforce Applications). You will find some of these benefits for developers, entrepreneurs, and CRM platform owners:

Benefits for developers

As a developer, you enjoy the following benefits for developing Salesforce Applications:

- **More time to focus on core development:** As a developer, you would like to spend more time building core business functionalities and less time setting up the development environment and tools that you need. You would like to use all your brain power to solve business problems and challenges rather than worry about getting a set of libraries to work with each other. The Salesforce platform comes with a plethora of low-code/no-code development tools and a simplified coding environment to build more complex code. These tools help you focus on developing core business functionalities away from being bothered about nitty-gritty detail but also enables you to deliver these functionalities much faster. Low-code development platforms can speed up the development process by up to ten times. You can read the *Forrester* report at the following link **https://www.forrester.com/blogs/why-you-need-to-know-about-low-code-even-if-youre-not-responsible-for-software-delivery/**
- **Market value:** Salesforce developers have been in high demand for over a decade. High demand drives high wages and offers more chances to progress in careers.

Several Salesforce developers decided to go solo and offer their services on a contract basis or as **Freelancers**.

- **The server is always up and running:** Developers, system administrators, infrastructure engineers, and network engineers always dread that moment when the server is down. Nobody likes to deal with this critical moment. The Salesforce platform is offered as a SaaS, meaning someone else is taking care of the server's uptime for you. Salesforce has one of the most impressive uptime numbers for any SaaS. This is not only due to the solid infrastructure used but to the reliable and stable building blocks offered to extend the standard functionalities and build Salesforce Applications. When you develop your Salesforce Application, you can focus on its functionalities and the value it adds. Salesforce will ensure that you are not bothered by uptime concerns.

- **Someone else is marketing it for you:** So, you have built your application; what is next? You need to market it and sell it. You likely need a dedicated team to do so. This means additional upfront costs and, therefore, more potential risk. The **Salesforce AppExchange** ensures your product is listed on the very-popular marketplace for Salesforce customers. The AppExchange is the first place for architects and decision makers of Salesforce customers to look for quick and relatively cheap ways to extend their Salesforce implementations with additional functionalities. Furthermore, the AppExchange handles the tricky part of deploying your application to the customer's **Salesforce Org** and even facilitates a smooth way to provide upgrades. In exchange for these services, Salesforce takes a certain percentage of your application revenue. However, it is easy to justify that cost considering the added value.

- **More room to become a Rockstar:** Some developers have different skills and capabilities. Some might be better at understanding business problems and solving problems than writing hard-core code. However, survivor is the best developer in the land of pure coding. In the world of Salesforce, there is more room for others. The definition of the word developer differs in Salesforce compared to pure-coding environments. A Salesforce developer does not necessarily need to write code. You will see several examples throughout this book. Hard-core coders are still very much in demand and appreciated in the Salesforce world. However, non-coders or less-capable-coders also got a great career opportunity in the Salesforce world.

- **It is easy to learn:** This is not a secret. Salesforce is one of the easiest platforms to learn. Partially because it is easy to configure and develop using point-and-click tools, but primarily because of the fantastic and free-to-use learning website **Trailhead,** Salesforce made it easier for the public to experience and learn their platforms via structured, well-defined, and practical training materials offered via Trailhead. Learners can easily spin off a **Trailhead Playground (TP)** environment and practice configuring and developing functionalities on the **Salesforce platform**. No upfront cost, and no credit card is needed. All