**Internship Training Report**

Salesforce CRM

**Submitted By-** Akash Mishra (1701227202)

of

8th Sem,CSE

**Under Supervision of:** Ajay Pulugundula Capgemini

##### (Duration: 28th January, 2021-16th April, 2021)



Department of Computer Science & Engineering

**C. V. RAMAN GLOBAL UNIVERSITY, BHUBANESWAR**

April, 2021

## Declaration

I hereby declare that the internship report entitled “CUSTOM SEARCH PAGE FOR MBX CORPORATION” is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which to substantial extent has been accepted for the award of any degree of the university or another institute of higher learning.

**Akash Mishra 1701227202**

**05-05-2021**

## Acknowledgement

It gives me immense pleasure to express my sincere gratitude to our faculty coordinator Prof. Monalisa Mishra for her support and advices to get and complete internship in the above said organization.

I extend my sincere thanks to our HOD Dr. Rojalina Priyadarshini for her immeasurable support throughout my internship.

I also like to acknowledge the contribution of other faculty members of the Department of CSE for their cooperation and kind assistance in successful completion of this internship.

May 2021 Akash Mishra(1701227202)

## Contents

[DECLARATION… 1](#_bookmark0)

[ACKNOWLEDGEMENT. 2](#_bookmark1)

[CONTENTS 3](#_bookmark2)

WEEKLY OVERVIEW… 4

[CHAPTER 1](#_bookmark3)

[INTRODUCTION](#_bookmark4)

[Salesforce 5](#_bookmark5)

[What is Salesforce Used For? 6](#_bookmark6)

[How Does Salesforce Work? 7](#_bookmark7)

[Salesforce Business Practices 8](#_bookmark8)

[Salesforce Mergers and Partnerships 8](#_bookmark9)

[CHAPTER 2](#_bookmark10)

[BOOTCAMP TRAINING 12](#_TOC_250004)

SQL 12

[HTML… 14](#_TOC_250003)

[CSS 15](#_TOC_250002)

[Bootstrap 16](#_TOC_250001)

[Angular 17](#_TOC_250000)

[CHAPTER](#_bookmark16) 3

Salesforce basic Administrator [18](#_bookmark11)

[Salesforce Lightning 20](#_bookmark12)

[Apex Language 2](#_bookmark13)2

[Visualforce 2](#_bookmark14)4

[Lightning 2](#_bookmark15)6

[CHAPTER](#_bookmark16) 4

[Salesforce Apex](#_bookmark17) 28

Apex Triggers. 30

[Apex Testing](#_bookmark18) 31

[CHAPTER](#_bookmark19) 5

[Aura Components](#_bookmark20) 33

[Lightning Web Components](#_bookmark21) 35

[Lightning Components vs. Lightning Web Components](#_bookmark22) 36

[CHAPTER](#_bookmark19) 6

[Project](#_bookmark20) 36

[Project Output](#_bookmark21) 38

[CHAPTER](#_bookmark23) 7

[L](#_bookmark24)[earnin](#_bookmark23)[g Outcomes… 4](#_bookmark24)3

Assessment Marks……………………………………………………………..46

Certificate………………………………………………………………….…….47

WEEKLY OVERVIEW OF INTERNSHIP ACTIVITIES

|  |  |  |  |
| --- | --- | --- | --- |
| **Week** | **Date** | **Day** | **Name of the Topic/Module Completed** |
| 1st week | 01.02.2021 | Monday | Soft Skills Part 1 |
| 02.02.2021 | Tuesday | RDBMS +Web Basics |
| 03.02.2021 | Wednesday | RDBMS +Web Basics |
| 04.02.2021 | Thursday | RDBMS +Web Basics |
| 05.02.2021 | Friday | RDBMS +Web Basics |
| 06.02.2021 | Saturday | RDBMS +Web Basics |
| 2nd week | 08.02.2021 | Monday | RDBMS +Web Basics |
| 09.02.2021 | Tuesday | RDBMS +Web Basics |
| 10.02.2021 | Wednesday | Core Java 8 Development Tools |
| 11.02.2021 | Thursday | Core Java 8 Development Tools |
| 12.02.2021 | Friday | Core Java 8 Development Tools |
| 13.02.2021 | Saturday | Core Java 8 Development Tools |
| 3rd week | 15.02.2021 | Monday | Core Java 8 Development Tools |
| 16.02.2021 | Tuesday | Bootstrap and Angular |
| 17.02.2021 | Wednesday | Bootstrap and Angular |
| 18.02.2021 | Thursday | Bootstrap and Angular |
| 19.02.2021 | Friday | Bootstrap and Angular |
| 20.02.2021 | Saturday | Bootstrap and Angular |
| 4th week | 22.02.2021 | Monday | Soft Skills Part 2 |
| 23.02.2021 | Tuesday | **Core Java 8+Bootstrap and Angular 6+ DBMS**  **+Web Basics Test** |
| 24.02.2021 | Wednesday | Salesforce Basic Administration |
| 25.02.2021 | Thursday | Salesforce Basic Administration |
| 26.02.2021 | Friday | Salesforce Basic Administration |
| 27.02.2021 | Saturday | Salesforce Basic Administration |

|  |  |  |  |
| --- | --- | --- | --- |
| 5th week | 01.03.2021 | Monday | Soft Skills Part 2 |
| 02.03.2021 | Tuesday | Salesforce Basic Administration |
| 03.03.2021 | Wednesday | Soft Skills Part 3 |
| 04.03.2021 | Thursday | Salesforce Development – App  Builder |
| 05.03.2021 | Friday | Sprint 1 |
| 06.03.2021 | Saturday | Sprint 1 |
| 6th week | 08.03.2021 | Monday | Sprint 1 |
| 09.03.2021 | Tuesday | Sprint 1 |
| 10.03.2021 | Wednesday | Sprint 1 |
| 11.03.2021 | Thursday | Project Evaluation |
| 12.03.2021 | Friday | **Soft Skills Part-4** |
| 13.03.2021 | Saturday |  |
| 7th week | 15.03.2021 | Monday | Platform Developer I |
| 16.03.2021 | Tuesday | Platform Developer I |
| 17.03.2021 | Wednesday | Platform Developer I |
| 19.03.2021 | Thursday | Platform Developer I |
| 20.03.2021 | Friday | Platform Developer I |
| 21.03.2021 | Saturday | Platform Developer I |
| 8th week | 22.03.2021 | Monday | Platform Developer I |
| 25.03.2021 | Tuesday | Advance Apex Coding |
| 26.03.2021 | Wednesday | Advance Apex Coding |
| 27.03.2021 | Thursday | LWC basics |
| 28.03.2021 | Friday | LWC basics |
| 01.04.2021 | Saturday | LWC basics |
| 9th week | 02.04.2021 | Monday | LWC basics |
| 03.04.2021 | Tuesday | LWC basics |
| 04.04.2021 | Wednesday | LWC basics |
| 05.04.2021 | Thursday | L1 Preparation |
| 08.04.2021 | Friday | L1 Preparation |
| 09.04.2021 | Saturday | L1 Test |
| 10th week | 10.04.2021 | Monday | Sprint 2 |
| 11.04.2021 | Tuesday | Sprint 2 |
| 12.04.2021 | Wednesday | Sprint 2 |
| 15.04.2021 | Thursday | Project Evaluation |
| 16.04.2021 | Friday | Project Evaluation |

Chapter 1

### INTRODUCTION

#### Salesforce



Salesforce is a cloud computing service as a software (SaaS) company that specializes in customer relationship management (CRM). Salesforce's services allow businesses to use cloud technology to better connect with customers, partners and potential customers. The software has become the number one for customer success and helps businesses track customer activity, market to customers and many more services.

Salesforce impressed investors recently by crushing third quarter estimates, [reporting a third](https://investor.salesforce.com/about-us/investor/investor-news/investor-news-details/2018/Salesforce-Announces-Record-Third-Quarter-Fiscal-2019-Results/default.aspx) [quarter revenue of $4.5 billion](https://investor.salesforce.com/about-us/investor/investor-news/investor-news-details/2018/Salesforce-Announces-Record-Third-Quarter-Fiscal-2019-Results/default.aspx) - up 33% year over year.

The software company has become very popular in recent years. TheStreet's founder Jim Cramer even dubs the service as a ['Cloud King'](https://www.thestreet.com/investing/jim-cramer-cloud-giants-better-than-fang-14581225) and has been very bullish on the stock - seemingly with good reason.

Founded in 1999 by a former Oracle ([**ORCL**](https://www.thestreet.com/quote/orcl)) - [Get Report](https://secure2.thestreet.com/cap/prm.do?OID=033365&ticker=ORCL) executive Marc Benioff, Dave Moellenhoff, Frank Dominguez and Parker Harris, Salesforce is one of the first global companies to successfully employ a cloud-based CRM software. Salesforce has been able to leverage cloud technology and build a variety of applications for businesses to better connect

to their customers and help give them key insights into their services through analytics and apps.

While their applications are vast, according to Salesforce, their CRM primarily focuses on helping companies with customer retention, keeping their customers happy, seeking out and executive customer acquisition, giving companies insights into their customers and much more.

So, how do companies actually use Salesforce?

### What is Salesforce Used For?



As of 2017, Salesforce [reportedly had 150,000 companies](https://www.salesinsideinc.com/services-details/salesforce-customers-list) using their software - among which include Amazon ([**AMZN**](https://www.thestreet.com/quote/amzn)) - [Get Report](https://secure2.thestreet.com/cap/prm.do?OID=033365&ticker=AMZN) , Adidas ([**ADDYY**](https://www.thestreet.com/quote/addyy)) , ADP ([**ADP**](https://www.thestreet.com/quote/adp)) - [Get Report](https://secure2.thestreet.com/cap/prm.do?OID=033365&ticker=ADP) , American Express ([**AXP**](https://www.thestreet.com/quote/axp)) - [Get Report](https://secure2.thestreet.com/cap/prm.do?OID=033365&ticker=AXP) and [many, many more.](https://www.salesforce.com/customer-success-stories/?d=7010M000000NsQp&!page=1)

Companies use Salesforce to understand their customers, connect with them on a variety of levels and help grow their customer base.

The cloud-based software allows companies to track (in real time) analytics, customer success and support, customer complaints and a variety of other CRM functions with the ease of cloud storage and access wherever the users are.

[According to the Salesforce website,](https://www.salesforce.com/form/sem/salesforce.jsp?gclid=EAIaIQobChMI5eLemrj63gIV04uzCh0IIwPgEAAYASAAEgKEyPD_BwE&d=70130000002IBx6&nc=7010M000001yBeP&DCMP=KNC-Google&ef_id=EAIaIQobChMI5eLemrj63gIV04uzCh0IIwPgEAAYASAAEgKEyPD_BwE%3AG%3As&s_kwcid=AL!4604!3!118174245539!e!!g!!what%2520is%2520salesforce) companies that use the software see average increases in a variety of areas, including an average 27% increase in sales revenues, 32% increase in lead conversion, 34% increase in customer satisfaction and a 56% faster deployment.

Because of its diverse selection of clouds and applications, Salesforce is also used by companies to assist with marketing, tracking sales and spending and analyzing performance. A variety of different clouds allow users to analyze various data, maintain communication forums with customers, implement sales strategies and more.

In essence, Salesforce is the one-stop-shop for businesses to manage, maintain, communicate with and grow their customer base and revenue streams.

### How Does Salesforce Work?

So, how does Salesforce practically work?

The company is a [service as a software (SaaS)](https://searchcloudcomputing.techtarget.com/definition/Software-as-a-Service) - which means it uses a cloud-computing, software distribution model that hosts applications and makes them available online.

Salesforce hosts numerous different cloud platforms that allow companies to interact with different data and service their customers in various capacities.

As of 2020, Salesforce has multiple different cloud platforms - a service cloud, marketing cloud, health cloud, app cloud, community cloud, analytics cloud, IoT cloud, Chatter cloud, commerce cloud, Heroku engagement cloud and more.

[According to the company,](https://www.youtube.com/watch?v=di6iwHhrH6s) Salesforce's sales cloud gives companies the ability to track contacts, opportunities and manage a team to increase sales. The service cloud allows companies to connect with customers and deliver premium customer service through showing customer activity and resolving issues. With their marketing cloud, Salesforce helps companies track customer journeys while providing multichannel marketing campaigns, while their community cloud allows companies to directly interact with their customers and allows their customers to interact with each other.

Additionally, Salesforce has been [implementing artificial intelligence (AI) into their Einstein](https://www.salesforce.com/products/sales-cloud/features/sales-cloud-einstein/?d=cta-body-promo-1) [platform,](https://www.salesforce.com/products/sales-cloud/features/sales-cloud-einstein/?d=cta-body-promo-1) which helps simplify the analytics workflow and produce more accurate forecasting, among other benefits.

Still, Salesforce's entire model supports customer relationship management (CRM). But, what is CRM?

###### Customer Relationship Management (CRM)

Customer Relationship Management (CRM) is "a technology for managing all your company's relationships and interactions with customers and potential customers. The goal is simple: Improve business relationships. A CRM system helps companies stay connected to customers, streamline processes, and improve profitability," according to Salesforce.

On a basic level, CRM helps companies understand customer needs and data and facilitates deeper customer development and retention.

###### Cloud-Based Software

Salesforce was the first to successfully implement a cloud-based CRM software. As a cloud- based service, Salesforce capitalizes on the benefit of data storage and accessibility from anywhere to draw a customer base.

Using a cloud-based software allows companies to track livedata, community discussions and manage analytics from anywhere - and with constantly updating speed and accuracy.

### Salesforce Business Practices

Salesforce created the 1-1-1 Model of Integrated Philanthropy, where companies contribute 1% equity, 1% of their product and 1% of employee hours back to the community. The service developed Salesforce.org to give to nonprofits and educational institutions, supporting them with technology, volunteering and grants, [according to their site.](https://www.salesforce.org/about-us/)

Companies currently using this 1% pledge model include DocuSign ([**DOCU**](https://www.thestreet.com/quote/docu)) - [Get Report](https://secure2.thestreet.com/cap/prm.do?OID=033365&ticker=DOCU) , SurveyMonkey ([**SVMK**](https://www.thestreet.com/quote/svmk)) - [Get Report ,](https://secure2.thestreet.com/cap/prm.do?OID=033365&ticker=SVMK) Conga and more.

In addition to their philanthropy, Salesforce offers guidelines for best customer service practices [on their site,](https://www.salesforce.com/hub/service/customer-service-best-practices/) including hiring the right people, managing customer expectations, focusing on first impressions, consistently collecting data, personalizing the experience, being where the customers are and maintaining focus.

### Salesforce Mergers and Partnerships

Salesforce's list of acquisitions seems to grow by the minute.

Most recently, it was announced in August of 2019 that they would be [purchasing](https://www.thestreet.com/investing/salesforce-to-buy-clicksoftware-for-1-35-billion-15050028) [ClickSoftware for $1.35 billion](https://www.thestreet.com/investing/salesforce-to-buy-clicksoftware-for-1-35-billion-15050028). Salesforce also announced a [$15.7 billion acquisition of](https://www.thestreet.com/investing/stocks/salesforce-swoops-to-buy-tableau-software-in-15-7-billion-data-analytics-deal-14985456) [Tableau Software Inc.,](https://www.thestreet.com/investing/stocks/salesforce-swoops-to-buy-tableau-software-in-15-7-billion-data-analytics-deal-14985456) a data analytics group, in June 2019 and acquired [MuleSoft for $6.5](https://www.thestreet.com/story/14529521/1/salesforce-acquires-mulesoft-in-6-5-billion-deal.html) [billion](https://www.thestreet.com/story/14529521/1/salesforce-acquires-mulesoft-in-6-5-billion-deal.html), announced in March of 2018.

Alongside the acquisition of MuleSoft, Salesforce has recently integrated MuleSoft's "Anypoint" platform (which is a app-connecting dashboard) into their cloud products, and also recently launched its new 'Integration Cloud,' which TheStreet reported is a "toolbox aimed at integrating business processes" and streamlining processes.

MuleSoft's CEO Greg Schott was very optimistic about the merger earlier this year.

"When [Salesforce co-CEO Marc Benioff] and I met back in February...he said we're starting to see digital transformation as an industry, and I said that we see ourselves as the engine of that industry," Schott said, TheStreet reported in September. "Putting the companies together felt like a really amazing strategic fit."

Other acquisitions have included Rebel, Zorap, CloudCraze LLC, Datorama and [50 others.](https://www.crunchbase.com/search/acquisitions/field/organizations/num_acquisitions/salesforce)

Notably, Salesforce partnered up with Facebook in 2014 to bring analytics to B2B marketers.

Additionally, Salesforce and Apple teamed up in 2018 to help improve apps for businesses. The [pair announced](https://www.apple.com/newsroom/2018/09/apple-and-salesforce-partner-to-help-redefine-customer-experiences-on-ios/) that the team "brings together the number one customer relationship management platform and iOS."

The Apple-Salesforce partnership will reportedly implement Apple features like Face ID, Business Chat, and the new-in-iOS-12 Siri Shortcuts to Salesforce.

##### Chapter 2

#### BOOTCAMP TRAINING

The training was designed in such a way that we were first trained on the prerequisites before learning salesforce technology. We were trained on HTML, CSS, BOOTSTRAP, Angular, Java, DBMS basics. And a weekly training on soft skills were also given to us to improve our skills in better communication and better comprehension.

The next 5-6 weeks were involved in giving us training on salesforce. This training comprises of theory sessions, practical sessions, doubt clearing sessions, mock interviews and project implementations on Salesforce Administrative as well as Salesforce Platform Developer and Integration.

The work hours were from 8:30A.M- 6:00P.M. The starting sessions before lunch break is all about learning on concepts and basics lead by virtual trainers and post lunch we had to do the lab work i.e. practical implementations on all the theory topics and complete the lab assignments which are always time bounded.

**STRUCTURED QUERY LANGUAGE**

SQL is a language to operate databases; it includes database creation, deletion, fetching rows, modifying rows, etc. SQL is an **ANSI** (American National Standards Institute) standard language, but there are many different versions of the SQL language.

#### What is SQL?

SQL is Structured Query Language, which is a computer language for storing, manipulating and retrieving data stored in a relational database.

SQL is the standard language for Relational Database System. All the Relational Database Management Systems (RDMS) like MySQL, MS Access, Oracle, Sybase, Informix, Postgres and SQL Server use SQL as their standard database language.

Also, they are using different dialects, such as −

* MS SQL Server using T-SQL,
* Oracle using PL/SQL,
* MS Access version of SQL is called JET SQL (native format) etc.

#### Why SQL?

SQL is widely popular because it offers the following advantages −

* Allows users to access data in the relational database management systems.
* Allows users to describe the data.
* Allows users to define the data in a database and manipulate that data.
* Allows to embed within other languages using SQL modules, libraries & pre- compilers.
* Allows users to create and drop databases and tables.
* Allows users to create view, stored procedure, functions in a database.
* Allows users to set permissions on tables, procedures and views.

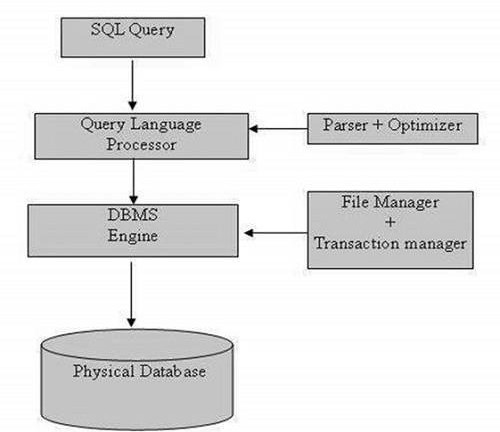
#### A Brief History of SQL

* **1970** − Dr. Edgar F. "Ted" Codd of IBM is known as the father of relational databases. He described a relational model for databases.
* **1974** − Structured Query Language appeared.
* **1978** − IBM worked to develop Codd's ideas and released a product named System/R.
* **1986** − IBM developed the first prototype of relational database and standardized by ANSI. The first relational database was released by Relational Software which later came to be known as Oracle.

#### SQL Process

When you are executing an SQL command for any RDBMS, the system determines the best way to carry out your request and SQL engine figures out how to interpret the task.

* Following is a simple diagram showing the SQL Architecture −



#### HTML

**HTML** stands for **Hyper Text Markup Language**, which is the most widely used language on Web to develop web pages. **HTML** was created by Berners-Lee in late 1991 but "HTML 2.0" was the first standard HTML specification which was published in 1995. HTML 4.01 was a major version of HTML and it was published in late 1999. Though HTML 4.01 version is widely used but currently we are having HTML-5 version which is an extension to HTML 4.01, and this version was published in 2012.

#### Why to Learn HTML?

Originally, **HTML** was developed with the intent of defining the structure of documents like headings, paragraphs, lists, and so forth to facilitate the sharing of scientific information between researchers. Now, HTML is being widely used to format web pages with the help of different tags available in HTML language.

**HTML** is a MUST for students and working professionals to become a great Software Engineer specially when they are working in Web Development Domain. I will list down some of the key advantages of learning HTML:

* **Create Web site** - You can create a website or customize an existing web template if you know HTML well.
* **Become a web designer** - If you want to start a carrer as a professional web designer, HTML and CSS designing is a must skill.
* **Understand web** - If you want to optimize your website, to boost its speed and performance, it is good to know HTML to yield best results.
* **Learn other languages** - Once you understands the basic of HTML then other related technologies like javascript, php, or angular are become easier to understand.

#### Applications of HTML

As mentioned before, HTML is one of the most widely used language over the web. I'm going to list few of them here:

* **Web pages development** - HTML is used to create pages which are rendered over the web. Almost every page of web is having html tags in it to render its details in browser.
* **Internet Navigation** - HTML provides tags which are used to navigate from one page to another and is heavily used in internet navigation.
* **Responsive UI** - HTML pages now-a-days works well on all platform, mobile, tabs, desktop or laptops owing to responsive design strategy.



* **Offline support** HTML pages once loaded can be made available offline on the machine without any need of internet.
* **Game development**- HTML5 has native support for rich experience and is now useful in gaming developent arena as well.

#### CSS

CSS is the language that defines the presentation of a web page. It is used to add color, background images, and textures, and to arrange elements on the page. However, CSS does a lot more than just paint a pretty picture. It is also used to enhance the usability of a website. The image below shows the front page of YouTube. On the left, is a regular rendering of the page, and on the right you can see how it looks without CSS. CSS stands for Cascading Style Sheets and it is the language used to style the visual presentation of web pages. CSS is the language that tells web browsers how to render the different parts of a web page.Every item or element on a web page is part of a document written in a markup language.

#### How is CSS Different From HTML?

The first thing to understand when approaching the topic of CSS is when to use a styling language like CSS and when to use a markup language such as HTML.

* All critical website *content* should be added to the website using a markup language such as HTML.
* *Presentation* of the website content should be defined by a styling language such as CSS.

Blog posts, page headings, video, audio, and pictures that are not part of the web page presentation should all be added to the web page with HTML. Background images and colors, borders, font size, typography, and the position of items on a web page should all be defined by CSS.

It’s important to make this distinction because failing to use the right language can make it difficult to make changes to the website in the future and create accessibility and usability issues for website visitors using a text-only browser or screen reader.



#### BOOTSTRAP

Bootstrap is a sleek, intuitive, and powerful, mobile first front-end framework for faster and easier web development. It uses HTML, CSS and Javascript. Bootstrap is a powerful toolkit - a collection of HTML, CSS, and JavaScript tools for creating and building web pages and web applications. It is a free and open source project, hosted on GitHub, and originally created by (and for) Twitter. With Bootstrap, [web developers](https://www.toptal.com/web) can concentrate on the development work, without worrying about design, and get a good looking website up and running quickly. Conversely, it gives web designers a solid foundation for creating interesting Bootstrap themes.

#### What Bootstrap Package Includes?

* **Scaffolding** − Bootstrap provides a basic structure with Grid System, link styles, and background. This is is covered in detail in the section **Bootstrap Basic Structure**
* **CSS** − Bootstrap comes with the feature of global CSS settings, fundamental HTML elements styled and enhanced with extensible classes, and an advanced grid system. This is covered in detail in the section **Bootstrap with CSS**.
* **Components** − Bootstrap contains over a dozen reusable components built to provide iconography, dropdowns, navigation, alerts, pop-overs, and much more. This is covered in detail in the section **Layout Components**.
* **JavaScript Plugins** − Bootstrap contains over a dozen custom jQuery plugins. You can easily include them all, or one by one. This is covered in details in the section **Bootstrap Plugins**.
* **Customize** − You can customize Bootstrap's components, LESS variables, and jQuery plugins to get your very own version.



#### ANGULAR

Angular is a platform and framework for building single-page client applications using HTML and TypeScript. Angular is written in TypeScript. It implements core and optional functionality as a set of TypeScript libraries that you import into your apps.The architecture of an Angular application relies on certain fundamental concepts. The basic building blocks of the Angular framework are Angular components that are organized into NgModules. NgModules collect related code into functional sets; an Angular app is defined by a set of NgModules. An app always has at least a root module that enables bootstrapping, and typically has many more feature modules.Components define views, which are sets of screen elements that Angular can choose among and modify according to your program logic and data.Components use services, which provide specific functionality not directly related to views. Service providers can be injected into components as dependencies, making your code modular, reusable, and efficient. Modules, components and services are classes that use decorators. These decorators mark their type and provide metadata that tells Angular how to use them.

The metadata for a component class associates it with a template that defines a view. A template combines ordinary HTML with Angular directives and binding markup that allow Angular to modify the HTML before rendering it for display.

The metadata for a service class provides the information Angular needs to make it available to components through dependency injection (DI).

An app's components typically define many views, arranged hierarchically. Angular provides the Router service to help you define navigation paths among views. The router provides sophisticated in-browser navigational capabilities.



#### CORE JAVA FUNDAMENTALS

Java programming language was originally developed by Sun Microsystems which was initiated by James Gosling and released in 1995 as core component of Sun Microsystems' Java platform (Java 1.0 [J2SE]).

The latest release of the Java Standard Edition is Java SE 8. With the advancement of Java and its widespread popularity, multiple configurations were built to suit various types of platforms. For example: J2EE for Enterprise Applications, J2ME for Mobile Applications.

The new J2 versions were renamed as Java SE, Java EE, and Java ME respectively. Java is guaranteed to be Write Once, Run Anywhere.

#### Java is −

Object Oriented − In Java, everything is an Object. Java can be easily extended since it is based on the Object model.

Platform Independent − Unlike many other programming languages including C and C++, when Java is compiled, it is not compiled into platform specific machine, rather into platform independent byte code. This byte code is distributed over the web and interpreted by the Virtual Machine (JVM) on whichever platform it is being run on.

Simple − Java is designed to be easy to learn. If you understand the basic concept of OOP Java, it would be easy to master.

Secure − With Java's secure feature it enables to develop virus-free, tamper-free systems. Authentication techniques are based on public-key encryption.

Architecture-neutral − Java compiler generates an architecture-neutral object file format, which makes the compiled code executable on many processors, with the presence of Java runtime system.

Portable − Being architecture-neutral and having no implementation dependent aspects of the specification makes Java portable. Compiler in Java is written in ANSI C with a clean portability boundary, which is a POSIX subset.

Robust − Java makes an effort to eliminate error prone situations by emphasizing mainly on compile time error checking and runtime checking.

Multithreaded − With Java's multithreaded feature it is possible to write programs that can perform many tasks simultaneously. This design feature allows the developers to construct interactive applications that can run smoothly.

Interpreted − Java byte code is translated on the fly to native machine instructions and is not stored anywhere. The development process is more rapid and analytical since the linking is an incremental and light-weight process.

High Performance − With the use of Just-In-Time compilers, Java enables high performance.

Distributed − Java is designed for the distributed environment of the internet.

Dynamic − Java is considered to be more dynamic than C or C++ since it is designed to adapt to an evolving environment. Java programs can carry extensive amount of run-time information that can be used to verify and resolve accesses to objects on run-time.

#### OOPS Concept

When we consider a Java program, it can be defined as a collection of objects that communicate via invoking each other's methods. Let us now briefly look into what do class, object, methods, and instance variables mean.

Object − Objects have states and behaviors. Example: A dog has states - color, name, breedas well as behavior such as wagging their tail, barking, eating. An object is an instance of a class.

Class − A class can be defined as a template/blueprint that describes the behavior/state that the object of its type supports.

Methods − A method is basically a behavior. A class can contain many methods. It is in methods where the logics are written, data is manipulated and all the actions are executed.

Instance Variables − Each object has its unique set of instance variables. An object's state is created by the values assigned to these instance variables.

#### Basic Syntax

About Java programs, it is very important to keep in mind the following points.

Case Sensitivity − Java is case sensitive, which means identifier Hello and hello would have different meaning in Java.

Class Names − For all class names the first letter should be in Upper Case. If several words are used to form a name of the class, each inner word's first letter should be in Upper Case.

Example: class MyFirstJavaClass

Method Names − All method names should start with a Lower Case letter. If several words are used to form the name of the method, then each inner word's first letter should be in Upper Case.

Example: public void myMethodName()

Program File Name − Name of the program file should exactly match the class name.

When saving the file, you should save it using the class name (Remember Java is case sensitive) and append '.java' to the end of the name (if the file name and the class name do not match, your program will not compile).

Example: Assume 'MyFirstJavaProgram' is the class name. Then the file should be saved as 'MyFirstJavaProgram.java'

public static void main(String args[]) − Java program processing starts from the main() method which is a mandatory part of every Java program.

Java Identifiers-All Java components require names. Names used for classes, variables, and methods are called identifiers.In Java, there are several points to remember about identifiers. They are as follows −All identifiers should begin with a letter (A to Z or a to z), currency character ($) or an underscore (\_).After the first character, identifiers can have any combination of characters.A key word cannot be used as an identifier.Most importantly, identifiers arecase sensitive.

Examples of legal identifiers: age, $salary, \_value, 1\_value. Examples of illegal identifiers: 123abc, -salary.

Java Modifiers -Like other languages, it is possible to modify classes, methods, etc., by using modifiers. There are two categories of modifiers −

Access Modifiers − default, public , protected, private;Non-access Modifiers − final, abstract, strictfp.We will be looking into more details about modifiers in the next section.Java VariablesFollowing are the types of variables in Java −Local Variables,Class Variables (Static Variables),Instance Variables (Non-static Variables)

Java Arrays -Arrays are objects that store multiple variables of the same type. However, an array itself is an object on the heap. We will look into how to declare, construct, and initialize in the upcoming chapters.

Java Enums -Enums were introduced in Java 5.0. Enums restrict a variable to have one of only a few predefined values. The values in this enumerated list are called enums.With the use of enums it is possible to reduce the number of bugs in your code.For example, if we consider an application for a fresh juice shop, it would be possible to restrict the glass size to small, medium, and large. This would make sure that it would not allow anyone to order any size other than small, medium, or large.

#### Example

Live Demo

class FreshJuice {

enum FreshJuiceSize{ SMALL, MEDIUM, LARGE } FreshJuiceSize size;

}

public class FreshJuiceTest {

public static void main(String args[]) { FreshJuice juice = new FreshJuice();

juice.size = FreshJuice.FreshJuiceSize.MEDIUM ; System.out.println("Size: " + juice.size);

}

}

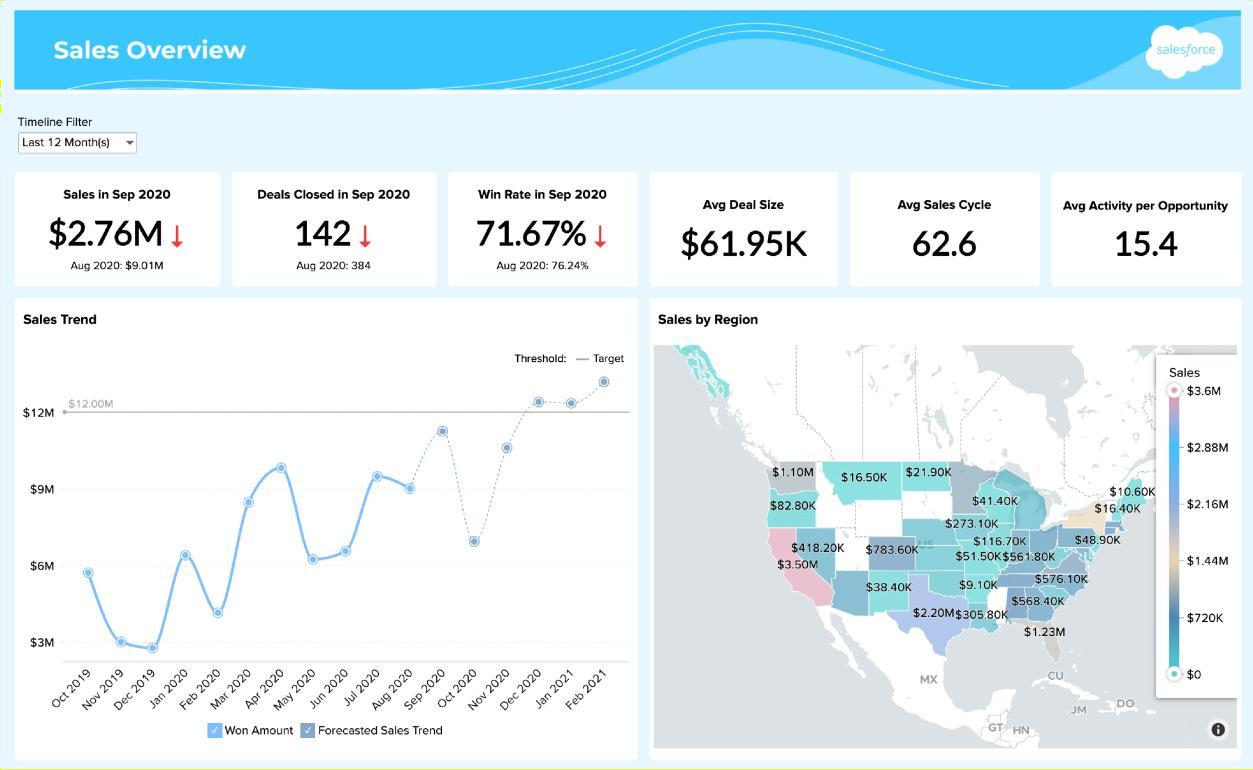
### Chapter 3

#### Salesforce Basic Administration

Salesforce is your customer success platform, designed to help you sell, service, market, analyze, and connect with your customers.Salesforce has everything you need to run your business from anywhere. Using standard products and features, you can manage relationships with prospects and customers, collaborate and engage with employees and partners, and store your data securely in the cloud.But standard products and features are only the beginning. Our platform allows you to customize and personalize the experience for your customers, partners, and employees and easily extend beyond out-of-the-box functionality.

The Service Cloud- Service Cloud customer service software gives you faster, smarter customer support. It refers to the “service” module in salesforce.com. It includes accounts, contacts, cases and solutions. It also has features like public knowledge base, web-to-case, call center and self service portal as well as customer service automation. Some of the related features are:

* Deliver 24/7 customer service
* Always-on customer service.
* Personalized service.
* Multichannel support.
* Intelligent customer service.
* Faster support.



#### Characteristics of Salesforce Solution

Salesforce is a SaaS product, but provides ability to extend and customize the application. Declarative (no development) coding allows for quick functionality deployment. Salesforce (system design) organization structures are foundational to the products and must be strategically and architecturally considered as part of any product purchase or deployment. Salesforce has a Platform as a Service solution,Force.com were other vendors can build proprietary solutions,AppExchange.Many of the AppExchange applications are very specialized for targeted/simple use cases, department or business unit vs Global.

#### Salesforce Cloud Tools

Accounts & contacts – Maximize sales rep productivity with a 360-degree view of each customer for deep knowledge of every account and contact.

Marketing & leads – See which of your marketing efforts leads to the most sales with a single system for managing and tracking multi-channel marketing campaigns from lead to close. Opportunities & quotes – Have a single place for updating deal information, recording customer interactions, tracking competitors, and creating quotes. Get at-a-glance visibility on that critical deal.

Visual process manager – Streamline pricing approvals to speed up your deal cycles and automate follow-up tasks to boost sales effectiveness.

Email & productivity – Remove the barriers to CRM by combining the desktop apps your reps already use in one spot. The Sales Cloud works seamlessly with Microsoft Office, Lotus Notes, and Google Apps.

Integrated content library – Give your reps instant, easy access to the best sales presentations and collateral. Share what works with the team and keep reps on message and selling with confidence.

Analytics – Get a comprehensive, real-time view of your business. Managers, executives, and reps are only a few clicks away from the insight needed to make smart business decisions and accurately estimate future sales.

Chatter – Collaborate on what matters most to you at work. Get updates on people, data, and documents through real-time feeds, where all the information you need is pushed straight to you.



#### What does a Salesforce Administrator do?

Salesforce Administrators work with stakeholders to define requirements and to customize the platform. To put it simply: they enable users to get the most out of Salesforce.

A Salesforce Admin best understands how to make the platform work for their company’s unique needs. Some companies may employ many people in this role. A Salesforce Administrator’s colleagues can rely on them to:

* + Maintain the platform
  + Make it as easy as possible for users of any technical level to use Salesforce
  + Stay updated on the platform’s new tools, capabilities, and updates

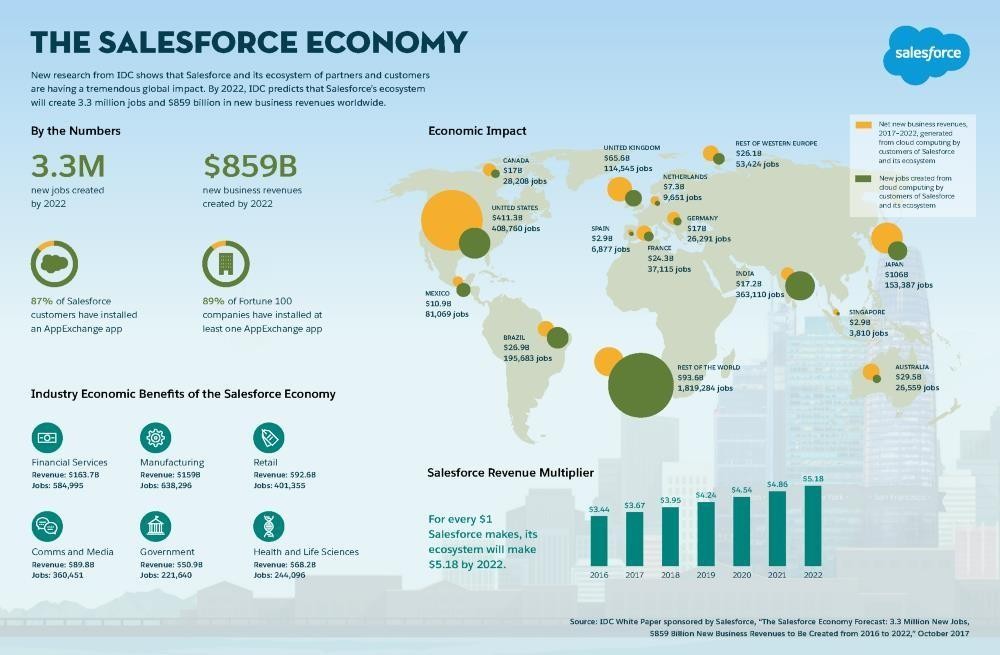
Think of Salesforce Administrators as your trusted advisors on all things Salesforce. They are a vital bridge between business and technology.

At some companies, these administrator jobs are combined with other roles. However, depending on their needs, some companies hire for this role specifically. Because of our platform’s capabilities, having someone with an admin certification can ensure the company is using the platform to its fullest potential.

Take, for example, your sales team. They might use Sales Cloud to keep track of opportunities and close deals. With the help of a Salesforce Admin, your salespeople can set up personalized dashboards, fields, alerts, and reports to shorten the sales cycle and track every lead in greater detail. This level of organization can lead to greater satisfaction among salespeople and new customers.

And that’s just one department. We have solutions for just about every team in a company, including sales, marketing, customer service, and more. Each department can learn from the Salesforce Admin how to use our platform most effectively. The concepts covered on the certifications are:

* + Managing users, data, and security
  + Maintaining and customizing [Sales Cloud](https://www.salesforce.com/products/sales-cloud/overview/) and [Service Cloud](https://www.salesforce.com/products/service-cloud/overview/) apps
  + Building reports, dashboards, and workflows



Salesforce Platform Developer



Salesforce Platform is the app development platform that extends your CRM’s reach and functionality. Force.com is a platform as a service (PaaS) that allows developers to create multitenant add-on applications that integrate into the main Salesforce.com application. Force.com applications are hosted on Salesforce.com's infrastructure. Force.com applications are built using Apex. Rapidly build enterprise mobile apps connected to your customers' data with Salesforce Platform Mobile Services. Leverage the tools, frameworks and APIs youneed to build apps for any device. Combine HTML5, native or hybrid apps with rich device features and your enterprise data to create engaging mobile apps.

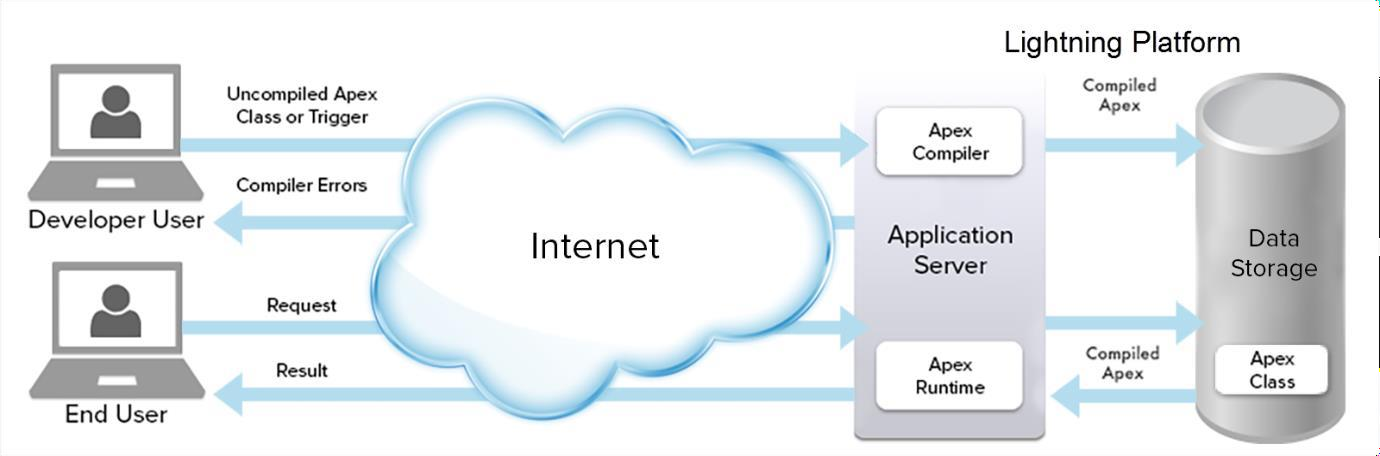
Salesforce Lightning



Get a completely re-imagined user interface with a seamless experience across all your devices. Build apps visually with Lightning App Builder and Lightning Components. Get tools and best practices with the Lightning Design System. Lightning Experience: The Lightning Experience brings a re-imagined consumer-like experience that is modern, efficient and smart to Salesforce users across every device (desktop, tablet and mobile). Relevant information is surfaced for each screen, streamlining processes and making workflows more intuitive. Lightning App Builder and Lightning Components: Instead of building applications from scratch, what ifyou could use a dragand-drop library of easily configurable components? With the Lightning App Builder and the Lightning Component Framework, developers and their business partners can combine custom and standard components with components from the AppExchange to build amazing apps even faster.

APEX LANGUAGE

Apex is a strongly typed, object-oriented programming language that allows developers to execute flow and transaction control statements on Salesforce servers in conjunction with calls to the API. Using syntax that looks like Java and acts like database stored procedures, Apex enables developers to add business logic to most system events, including button clicks, related record updates, and Visualforce pages. Apex code can be initiated by Web service requests and from triggers on objects.



#### Visual Force

Rapidly build enterprise mobile apps connected to your customers' data with Salesforce Platform Mobile Services. Leverage the tools, frameworks and APIs you need to build apps for any device. Combine HTML5, native or hybrid apps with rich device features and your enterprise data to create engaging mobile apps. Visualforce offers a user-friendly, split-screen development environment. Simply click Save to immediately test your code! The inline editor also includes auto-completion and limited syntax highlighting, as well as online documentation. Access to the inline editor is available only in Development Mode. Visualforce pages can also be edited through the Force.com IDE.

#### Visualforce or Lightning?



Being a framework, which comprises of a tag-based markup language and a set of server-side controllers, Visualforce enables developers to build custom user interfaces with a native hosting on the [Force.com](http://force.com/) platform. Lightning, on the other hand, is a modern UI framework used in the development of dynamic web apps for desktop and mobile devices. As both of them provide ways to create custom user interfaces for Salesforce, this gives rise to a comparison between Visualforce and Lightning. So which one should you choose?

#### Visualforce

With a robust set of tags resolved at the server-side, Visualforce framework is known as a page- centric web application model. Since VF relies on the server to generate a new page when a user interacts with the application, it becomes quite challenging to deliver a new and dynamic experience as expected by the users. However, it is considered great for its basic functionalities. Visualforce uses a markup language similar to HTML for designing pages and Apex code for handling database operations. Let us take a look at the classification of Visualforce pages:

* + Visualforce Page– A markup language like HTML is used for designing the layout of pages.
  + Custom Controller– Apex code is used to handle backend processing (server-side operations) implemented at the VF page.
  + JavaScript– JavaScript is optional and helps in managing client-side processing and can be coupled with CSS for a revamp of the interface.
  + Apex Extensions– Logical operations can be performed using Apex extensions which are not available in standard controllers. However, this is optional too.

#### Lightning

With JavaScript being used at the client-side as well as the involvement of Apex on the server- side, Lightning framework is known as a new app-centric model that provides business and data logics to develop dynamic web applications meant for desktop and mobile devices. This framework does not entirely replace a page at a time, rather, it leverages JavaScript to create, transform, and animate the user interface. Lightning Components and Applications uses “Bundles” to store necessary files and ensures the fluid motion of user interface. Here’s what a Lightning Component Bundle includes:

* + Component– A markup language is used for configuring the layout of the component.
  + Controller– A JavaScript Controller is used for handling client-side processing or an apex controller is preferred for server-side processing.
  + Design File– A design file is used for describing the design-time behavior of the component being used in Lightning pages or in Lightning App Builder.
  + Documentation File– A documentation file is used to provide reference documents or sample code to users who have received your component.
  + Helper– A helper aims at storing reusable JavaScript functions which are handled by the controller.
  + SVG File– An SVG File enables you to include custom icons that can be used for reference further.

##### Chapter 4

Salesforce Apex



**Apex** is an object-oriented and strongly typed programming language developed by Salesforce for building Software as a Service (SaaS) and Customer Relationship Management (CRM). Apex helps developers to create third-party SaaS applications and add business logic to system events by providing back-end database support and client-server interfaces. Apex helps developers to add business logic to the system events like button clicks, related record updates, and Visualforce pages. Apex has a similar syntax to Java. Developer Action: All the apex code written by a developer is compiled into a set of instructions that can be understood by apex runtime interpreter when the developer saves the code to the platform and these instructions then save as metadata to the platform.End User Action: When the user event executes an apex code, the platform server gets the compiled instructions from metadata and runs them through the apex interpreter before returning the result.

As a language, Apex is:

Integrated

Apex provides built-in support for common Lightning Platform idioms, including:

* + - Data manipulation language (DML) calls, such as INSERT, UPDATE, and DELETE, that include built-in DmlException handling
    - Inline Salesforce Object Query Language (SOQL) and Salesforce Object Search Language (SOSL) queries that return lists of sObject records
    - Looping that allows for bulk processing of multiple records at a time
    - Locking syntax that prevents record update conflicts
    - Custom public API calls that can be built from stored Apex methods
    - Warnings and errors issued when a user tries to edit or delete a custom object or field that is referenced by Apex

Easy to use

Apex is based on familiar Java idioms, such as variable and expression syntax, block and conditional statement syntax, loop syntax, object and array notation. Where Apex introduces new elements, it uses syntax and semantics that are easy to understand and encourage efficient use of the Lightning Platform. Therefore, Apex produces code that is both succinct and easy to write.

Data focused

Apex is designed to thread together multiple query and DML statements into a single unit of work on the Salesforce server. Developers use database stored procedures to thread together multiple transaction statements on a database server in a similar way. Like other database stored procedures, Apex does not attempt to provide general support for rendering elements in the user interface.

Rigorous

Apex is a strongly typed language that uses direct references to schema objects such as object and field names. It fails quickly at compile time if any references are invalid. It stores all custom field, object, and class dependencies in metadata to ensure that they are not deleted while required by active Apex code.

Hosted

Apex is interpreted, executed, and controlled entirely by the Lightning Platform.

Multitenant aware

Like the rest of the Lightning Platform, Apex runs in a multitenant environment. So, the Apex runtime engine is designed to guard closely against runaway code, preventing it from monopolizing shared resources. Any code that violates limits fails with easy-to- understand error messages.

Easy to test

Apex provides built-in support for unit test creation and execution. It includes test results that indicate how much code is covered, and which parts of your code could be more efficient. Salesforce ensures that all custom Apex code works as expected by executing all unit tests prior to any platform upgrades.

Versioned

You can save your Apex code against different versions of the API. This enables you to maintain behavior.

Apex is included in Performance Edition, Unlimited Edition, Developer Edition, Enterprise Edition, and Database.com.

Features of Apex Programming Language

* + Apex is a case insensitive language.
  + You can perform DML operations like INSERT, UPDATE, UPSERT, DELETE on sObject records using apex.
  + You can query sObject records using SOQL(salesforce object query language) and SOSL(salesforce object search language) in apex.
  + Allows you to create a [unit test](https://www.guru99.com/unit-testing-guide.html) and execute them to verify the [code coverage](https://www.guru99.com/code-coverage.html) and efficiency of the code in apex.
  + Apex executes in a multi-tenant environment, and [Salesforce](https://www.guru99.com/salesforce-tutorial.html) has defined some governor limits that prevent a user from controlling the shared resources. Any code that crosses the salesforce governor limit fails, an error shows up.
  + Salesforce object can be used as a datatype in apex. For example - Account acc = new Account();
  + Apex automatically upgrades with every Salesforce release.

#### Apex Trigger



Apex triggers enable you to execute custom apex before and after a DML operation is performed.Apex support following two types of triggers:

Before triggers: These triggers are used to validate and update the field's value before the record save to the database.

After triggers: These triggers are used to access the fields(record ID, LastModifiedDate field) set by the system after a record committed to the database. These fields value can be used to modify other records. Records that fires after triggers are read-only.

It is a best practice to write bulky triggers. A bulky trigger can process a single record as well as multiple records at a time.Syntax of an apex trigger:

trigger TriggerName on ObjectName (trigger\_events) {

//Code\_block

}

Here, TriggerName is the name of the trigger, ObjectName is the name of the object on which

trigger to be written, trigger\_events is the comma-separated list of events. Following are the events supported by the apex triggers: before insert, before the update, before delete, after insert, after an update, after delete, after undelete. Static keywords can't be used in an Apex trigger. All the keywords applicable to inner classes can be used in an Apex trigger. T here are implicit variable define by every trigger that returns the run-time context. These variables are defined in the system. Trigger class. These variables are called context variables.

#### Apex Testing

Testing is the integrated part of Apex or any other application development. In Apex, we have separate test classes to develop for all the unit testing. In SFDC, the code must have 75% code coverage in order to be deployed to Production. This code coverage is performed by the test classes. Test classes are the code snippets which test the functionality of other Apex class. Let us write a test class for one of our codes which we have written previously. We will write test class to cover our Trigger and Helper class code. Below is the trigger and helper class which needs to be covered.



trigger Customer\_After\_Insert on APEX\_Customer c (after update) { CustomerTriggerHelper.createInvoiceRecords(Trigger.new, trigger.oldMap);

//Trigger calls the helper class and does not have any code in Trigger

}

// Helper Class:

public class CustomerTriggerHelper {

public static void createInvoiceRecords (List<apex\_customer c>

customerList, Map<id, apex\_customer c> oldMapCustomer) { List<apex\_invoice c> InvoiceList = new List<apex\_invoice c>();

for (APEX\_Customer c objCustomer: customerList) {

if (objCustomer.APEX\_Customer\_Status c == 'Active' && oldMapCustomer.get(objCustomer.id).APEX\_Customer\_Status c == 'Inactive') {

// condition to check the old value and new value APEX\_Invoice c objInvoice = new APEX\_Invoice c();

objInvoice.APEX\_Status c = 'Pending'; objInvoice.APEX\_Customer c = objCustomer.id; InvoiceList.add(objInvoice);

}

}

insert InvoiceList; // DML to insert the Invoice List in SFDC

}

}

##### Chapter 5

Aura Components

Aura is a UI framework for developing dynamic web apps for mobile and desktop devices.Aura supports partitioned multi-tier component development that bridges the client and server. It uses JavaScript on the client side and Java on the server side. Aura components are the self- contained and reusable units of an app. They represent a reusable section of the UI, and can range in granularity from a single line of text to an entire app.The framework includes a set of prebuilt components. For example, components that come with the Lightning Design System styling are available in the lightning namespace. These components are also known as the base Lightning components. You can assemble and configure components to form new components in an app. Components are rendered to produce HTML DOM elements within the browser.A component can contain other components, as well as HTML, CSS, JavaScript, or any other Web-enabled code. This enables you to build apps with sophisticated UIs.The details of a component's implementation are encapsulated. This allows the consumer of a component to focus on building their app, while the component author can innovate and make changes without breaking consumers. You configure components by setting the named attributes that they expose in their definition. Components interact with their environment by listening to or publishing events.

Why Use the Aura Components Programming Model?

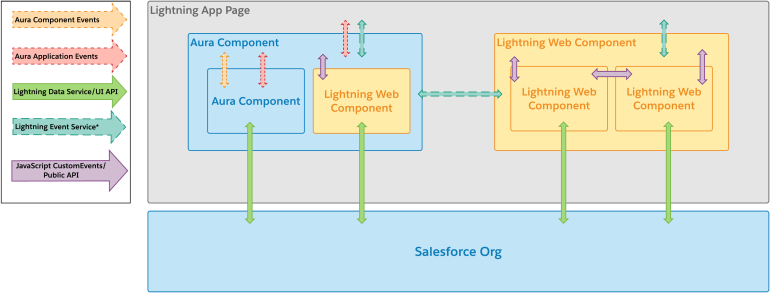
The benefits include an out-of-the-box set of components, event-driven architecture, and a framework optimized for performance.

Out-of-the-box Components -Comes with an out-of-the-box set of components to kick start building apps. You don't have to spend your time optimizing your apps for different devices as the components take care of that for you.

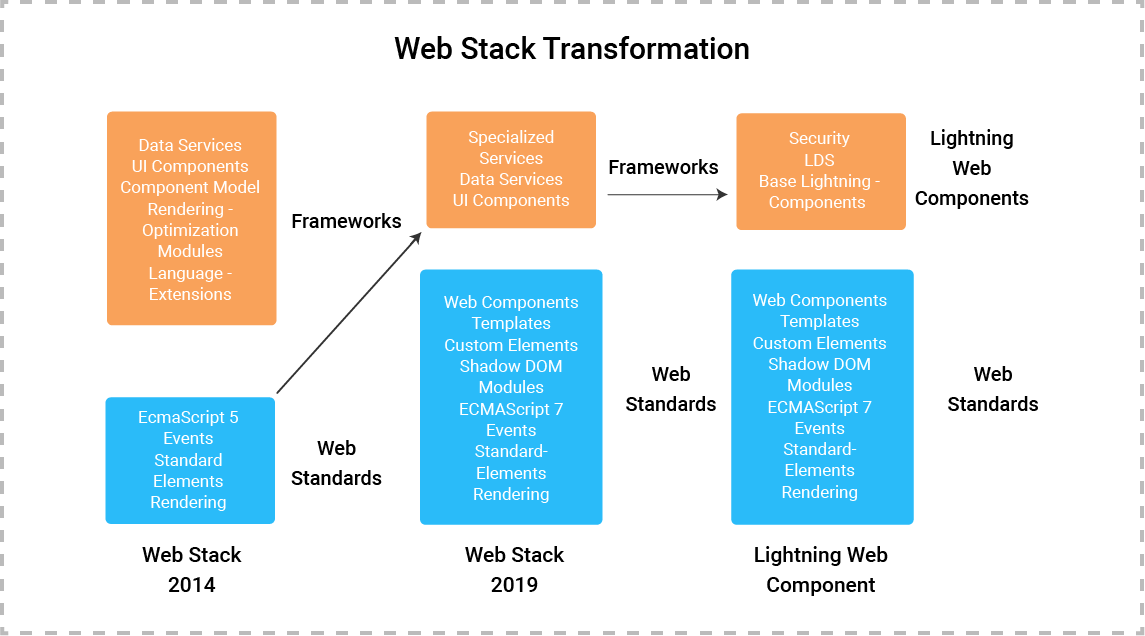
Rich Custom Component Ecosystem - Create business-ready components and make them available in the Salesforce mobile app, Lightning Experience, and Experience Builder sites. Salesforce mobile app users access your components via the navigation menu. Customize Lightning Experience or create your own Lightning pages using drag-and-drop components in the Lightning App Builder. Create and customize Experience Builder sites using Experience Builder. Additional components are available for your org in the AppExchange. Similarly, you can publish your components and share them with other users.

Fast Development - Empowers teams to work faster with out-of-the-box components that function seamlessly with desktop and mobile devices. Building an app with components facilitates parallel design, improving overall development efficiency.Components are encapsulated and their internals stay private, while their public shape is visible to consumers of the component. This strong separation gives component authors freedom to change the internal implementation details and insulates component consumers from those changes.

Device-aware and Cross Browser Compatibility -Apps use responsive design and support the latest in browser technology such as HTML5, CSS3, and touch events.



Lightning Web Components



Lightning Web Components are an updated web standards-based framework method for creating lightning components on the Salesforce Platform, Lightning Web Components utilize standard tech like CSS, HTML, and updated JavaScript without requiring a set framework, incorporating the latest innovations in JavaScript, including Shadow Document Object Model(DOM), custom elements, and web components (ECMAScript 7 is specifically the updated JavaScript language used). This is what gives them an edge over Lightning Components – they are simpler to define and develop, greatly because they are in-line with updated web standards. This means that developers with skills in these technologies can easily use this framework to develop Lightning Components for their pages, plus users don’t have to be as familiar with the Salesforce Platform to deploy these components (as is required with regular Lightning Components). Overall, this results in more people building Lightning Components more efficiently, effectively opening the doors for admins to utilize a wider range of professionals to get the help they need.

Benefits of Lightning Web Components

Lightning Web Components take many of the benefits from regular Lightning Components and amplifies several areas.

For Developers

* + Standardized-based architecture, can build components for pages faster.
  + Code reuse is supported for quicker deployments.
  + Support unit testing for quality assurance.
  + Access to more developer talent who can quickly use Lightning Web Components to create desired solutions.

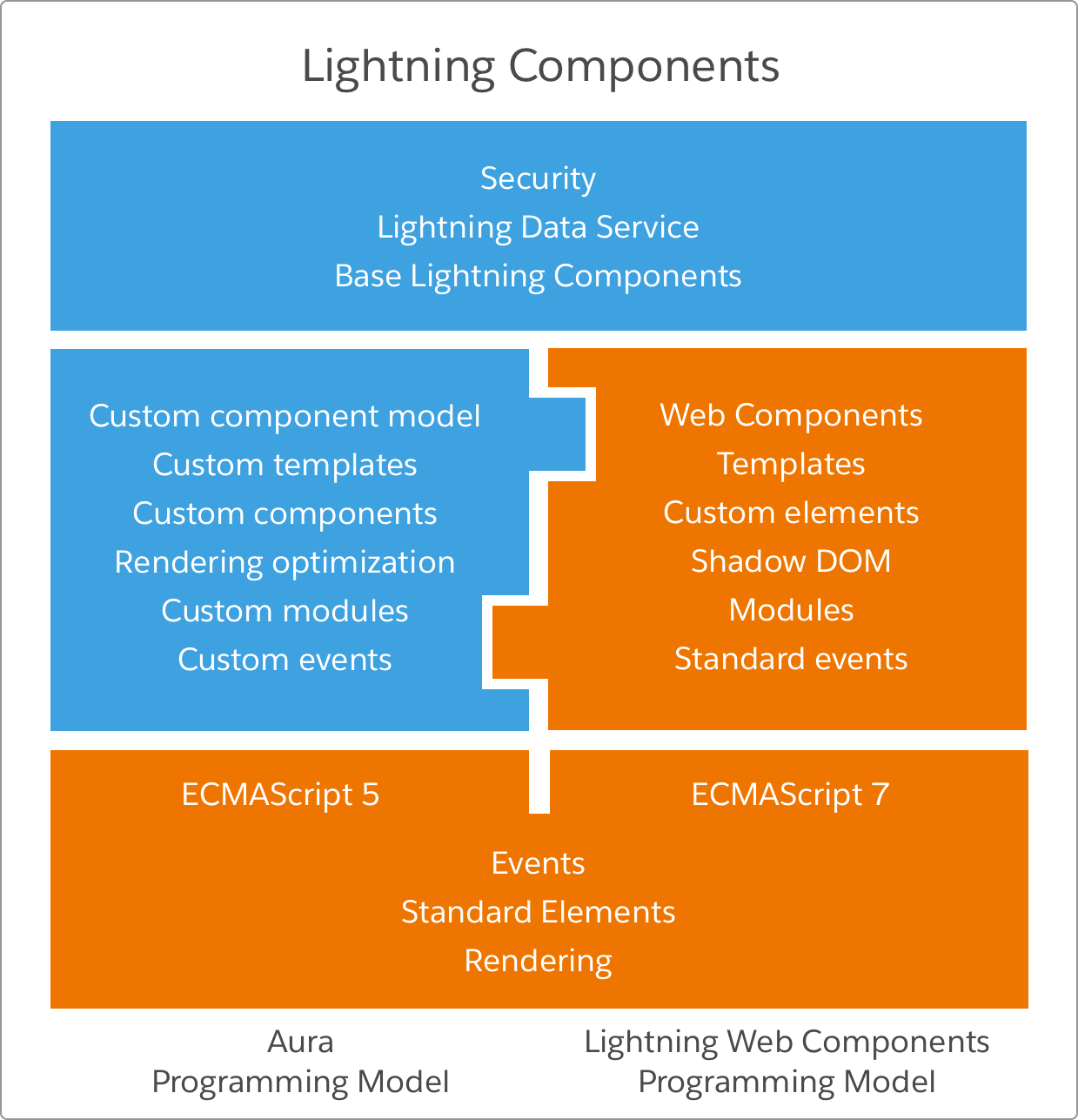
For Admins

* + The advanced standard-based architecture, Lightning Web Components make the process of building a Lightning page much easier.
  + Implement Lightning Web Components with just clicks in the application.

For Users

* + The components run more fluidly without browser interferences, enhancing performance.

Lightning Components vs. Lightning Web Components:



Chapter 6

**Project**

**Introduction**

MBX Corporation has implemented a Sales application in order to manage daily sales operations for their Accounts/Customers. A Sales User would like to see a custom UI for Opportunity search on the ‘Sales’ App which will provide them view to Opportunity matching the search criteria and sending them over to 3rd party application. User Adoption for the new page can be measured on the Report & Dashboard which will be showing the Integration Status for Opportunities.

Use Case

* Create custom search page on Opportunity object in Salesforce classic (with sidebar) or Lightening UI containing a Text box. (VF/LWC)
* User enters search criteria which is matched with below mentioned fields:

1. Name
2. Account Name
3. Stage
4. Type
5. Amount.

* Add above mentioned fields on page as part of each column on the search table. (VF/LWC)
* Create 2 new fields on Opportunity object mentioned below:

1. Integration Status – Text (50 char)
2. Integration Comments – Text (200 char)

* ‘Name’ & ‘Account Name’ column will be containing a link to each Opportunity/Account record page which opens in a new tab. (VF/LWC)
* Add pagination to the custom page & display 20 records on a page after the Search. (VF/LWC)
* Create a custom validation on Opportunity to allow users setting Closed Date only if Integration

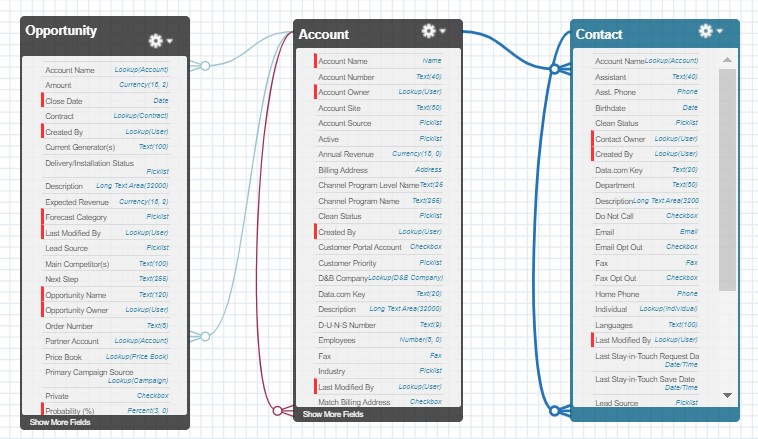
Status value is ‘Success’ (Validation Rule/Apex Trigger)

* Provide a custom button/link/action on each search row to send your Opportunity to 3rd party application. (Apex Class/Apex REST callout)
* Create a new Controller & add it on the page to perform Search operation and sending out HTTP request. (Apex Class/Apex REST callout)
* Check the response from HTTP request, if the Status Code is 200 then update Integration Status & Comments with ‘Success’ value on the Opportunity. In case of any other Status code store the error information on same fields. (Apex Class/Apex REST callout)
* Create a Test class providing more than 75% code coverage for above mentioned Controller Class. (Apex Test Class)
* Create a class using Mock interface and use it on previous Apex Test class to cover more than 75% code coverage on REST callout. (Apex Test Class)
* Display the error message on custom search page when either no records meet the criteria or HTTP request fails. (Apex class/VF/LWC)
* Create a new Tab to launch the custom search page and add it to the ‘Sales’ application. (Custom Tab/Lightening App)

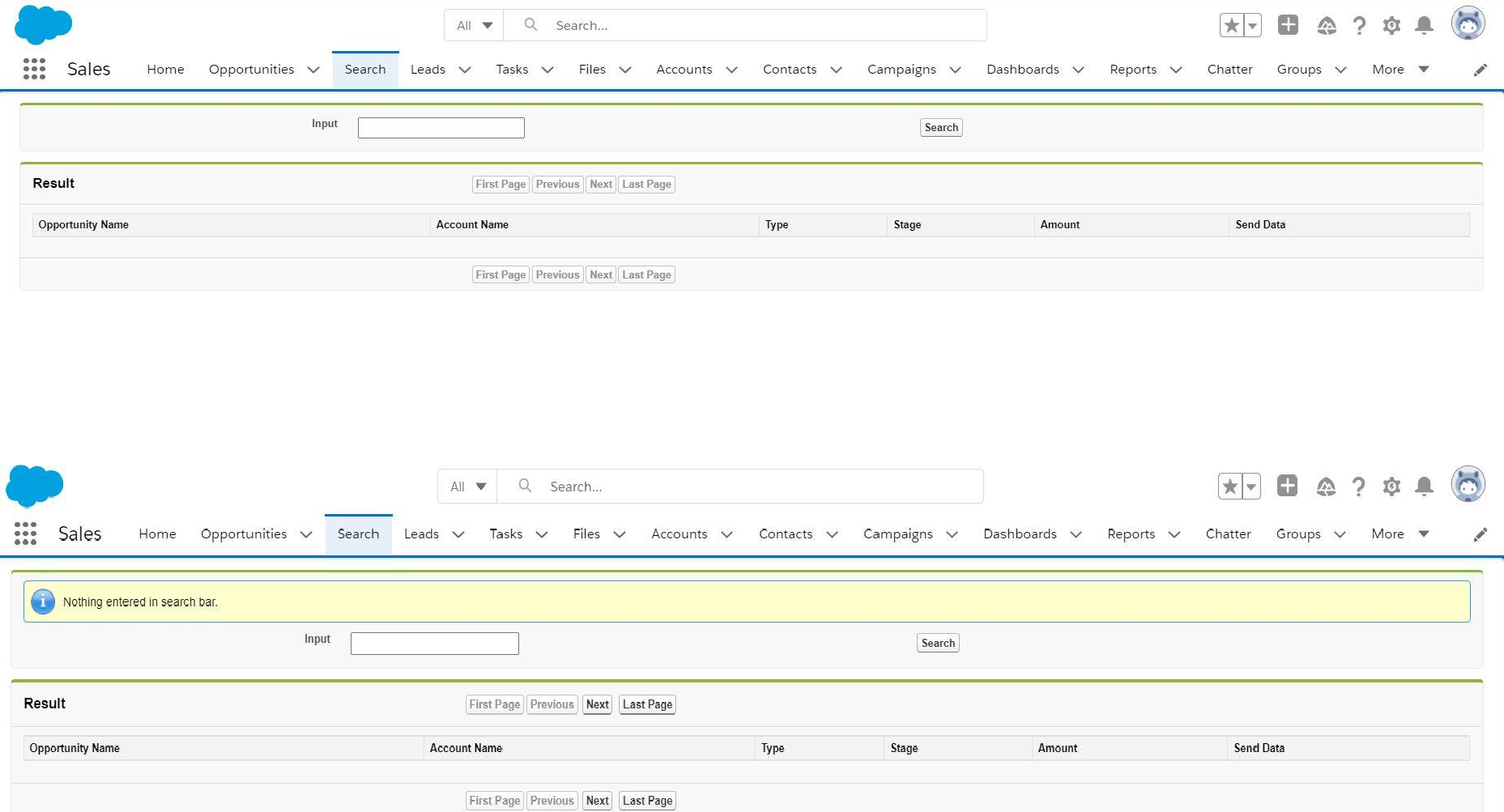
Report & Dashboard

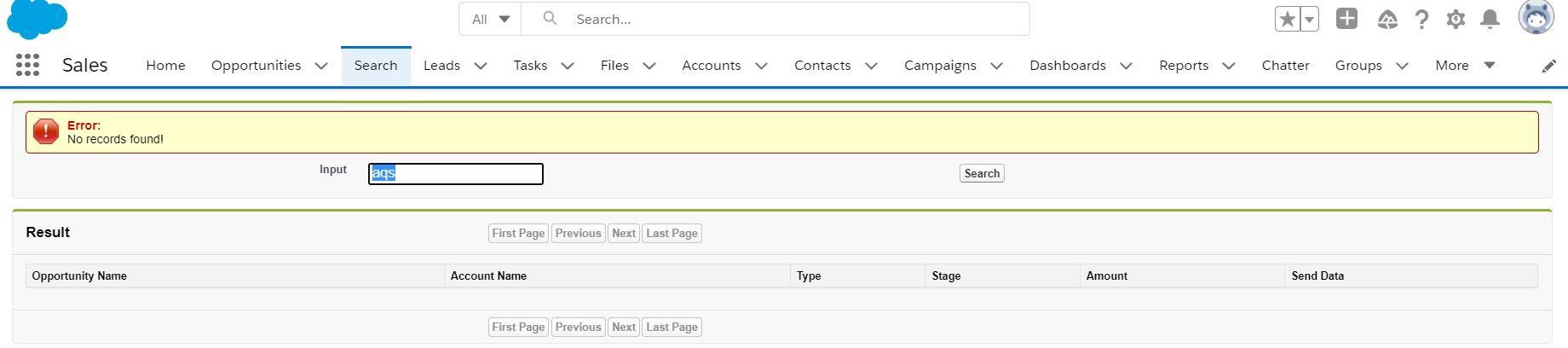
* Create a Report on Opportunity object to see all the records successfully sent over to 3rd party app.
* Display a Dashboard based on the previous Report to show Opportunity with successful integration.

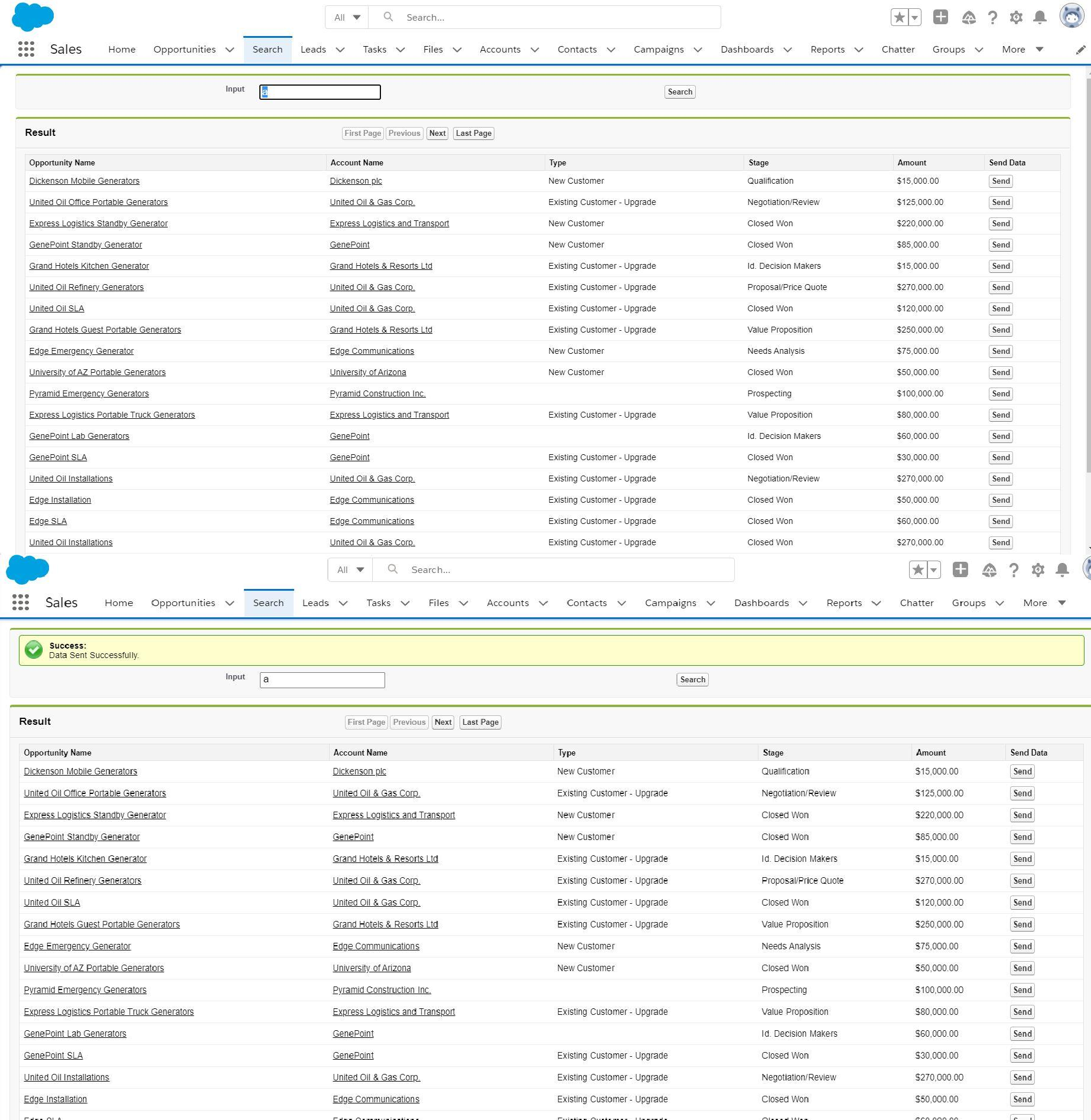
Object ER Diagram

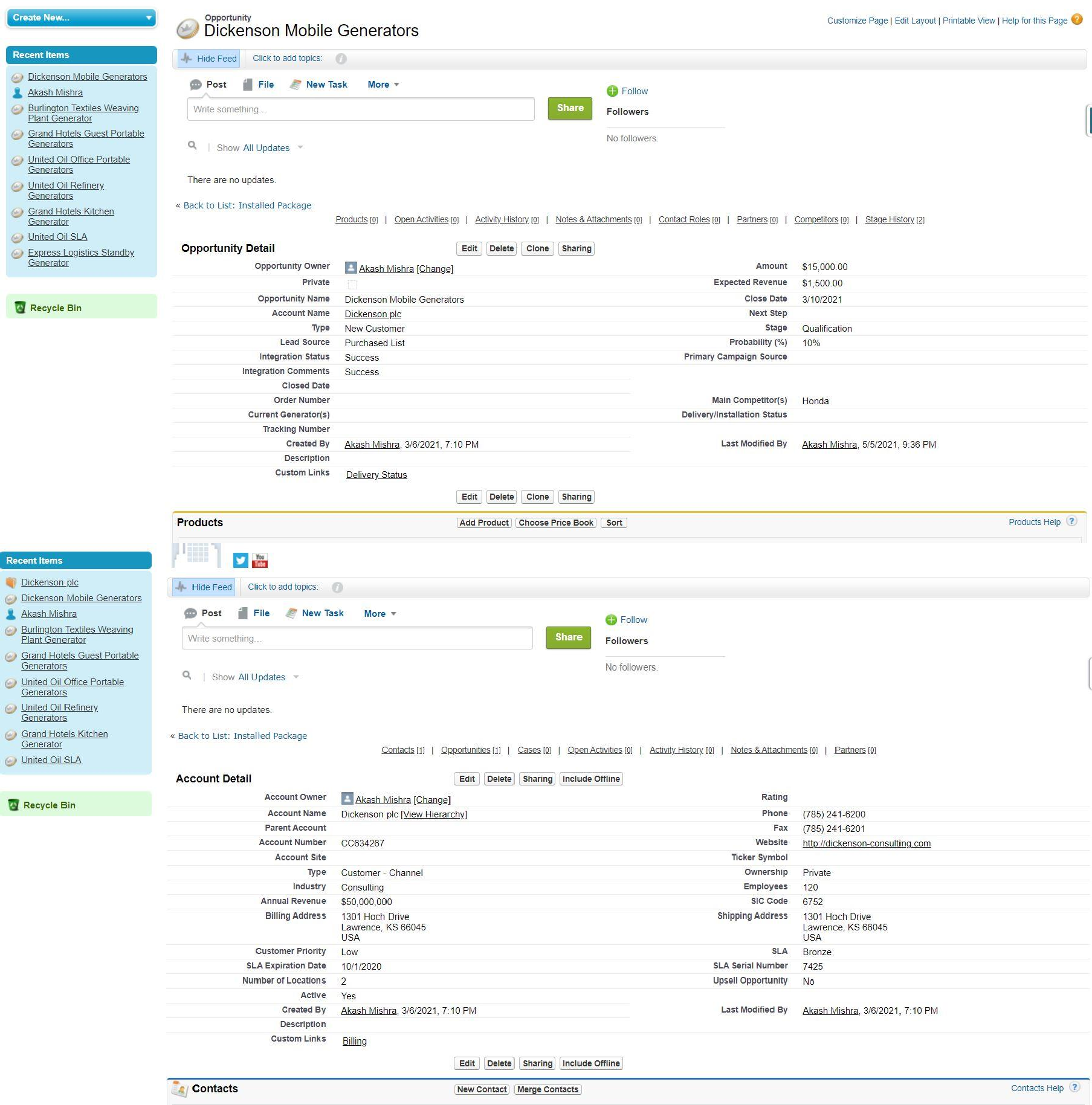


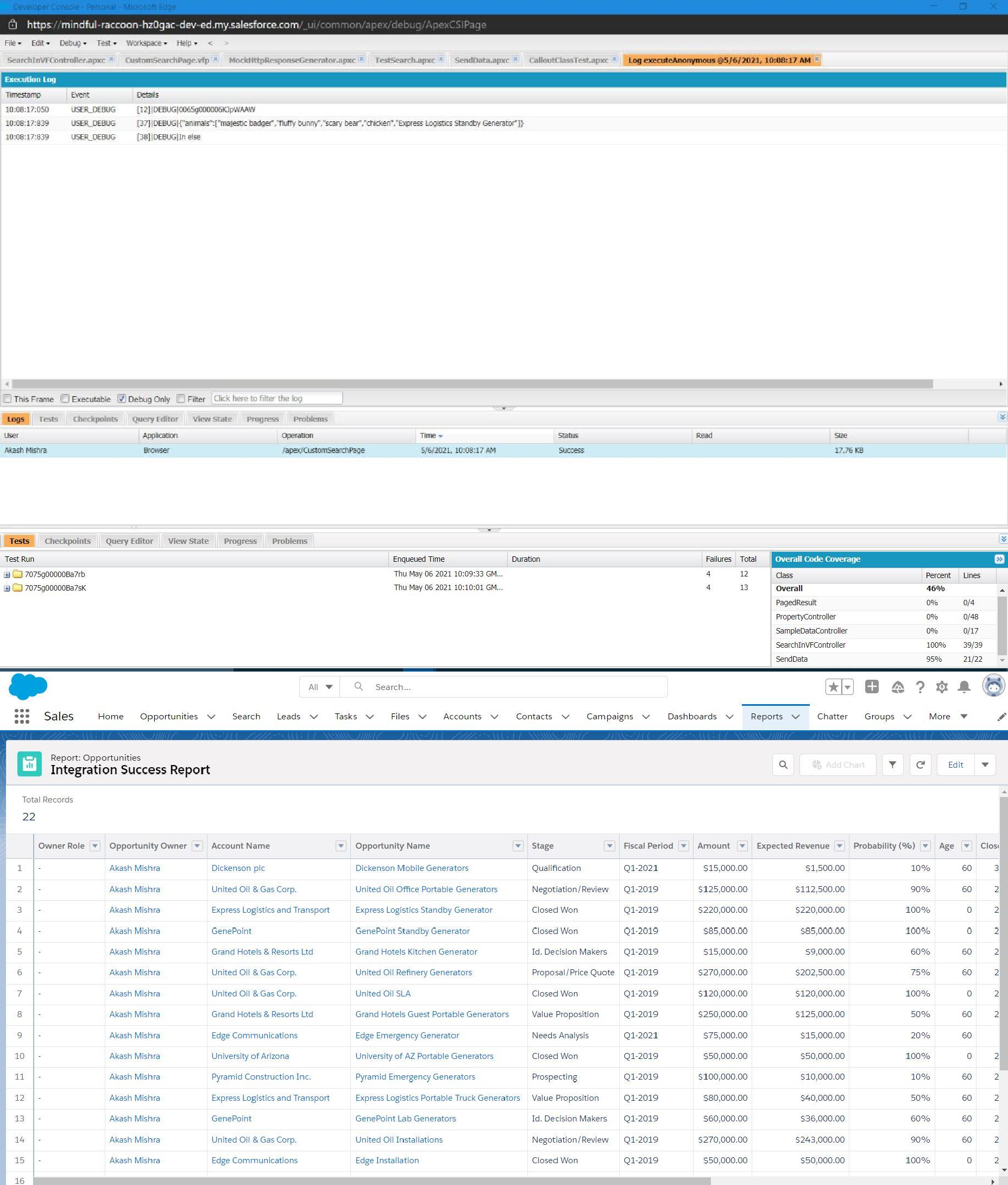
**Project Output**

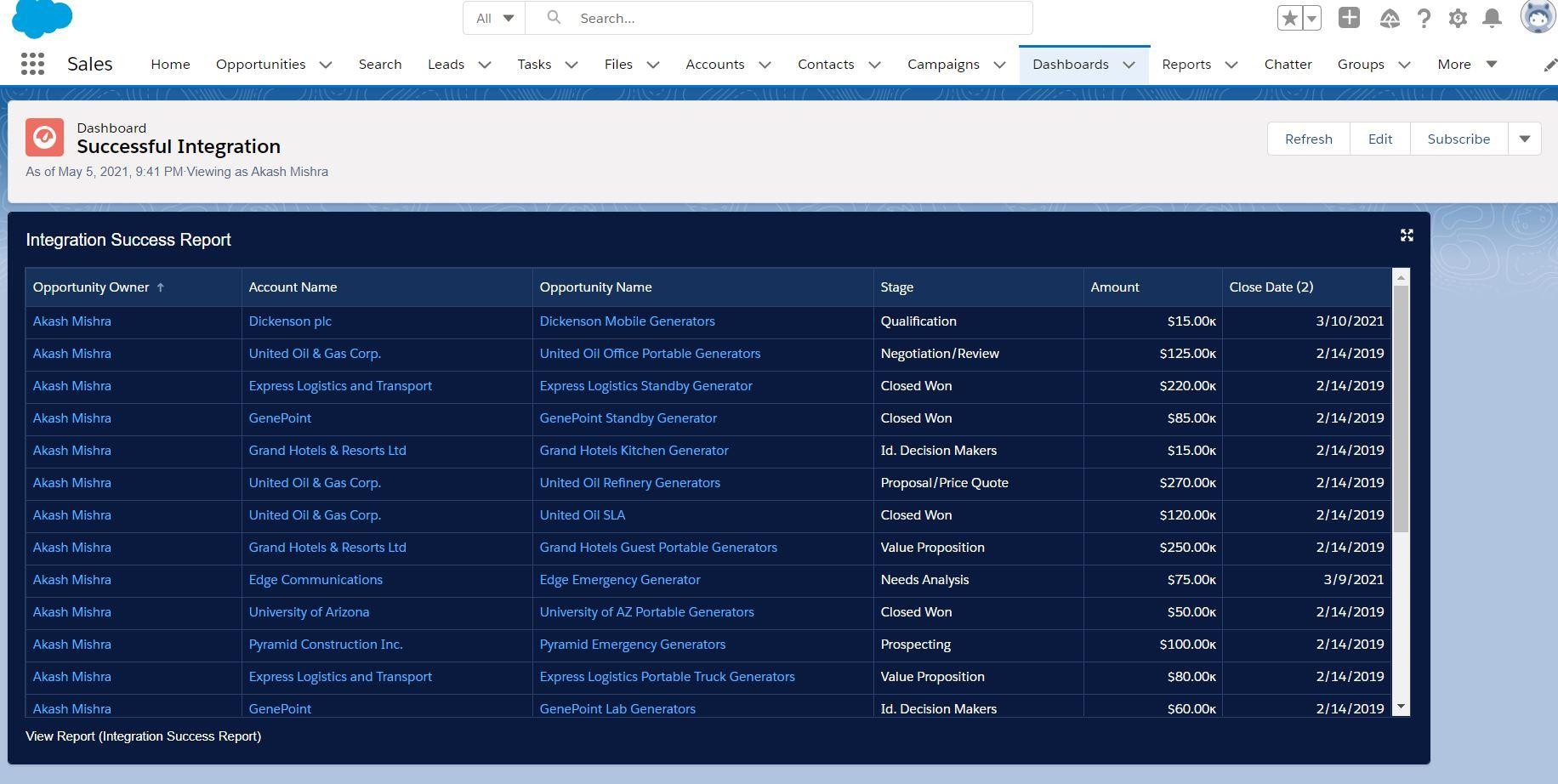












Chapter 7

# Learning Outcomes

After the successful completion of module 3 of the internship program we were able to do the following tasks:

SALESFORCE FUNDAMENTALS

* + Describe the considerations when developing in a multi-tenant environment.
  + Describe how the Salesforce platform features map to the MVC pattern.
  + Describe the capabilities of the core CRM objects in the Salesforce schema.
  + Identify the common scenarios for extending an application's capabilities using the AppExchange.
  + Identify common use cases for declarative customization of the Lightning Platform, and customization and features of the Heroku platform

DATA MODELING AND MANAGEMENT

* + Given a set of requirements, determine the appropriate data model.
  + Describe the capabilities of the various relationship types and the implications of each on record access, user interface (UI), and object-oriented programming.
  + Describe the impact of schema design and modifications on Apex Development.
  + Describe how to visualize and create entity relationships.
  + Describe the options for and considerations when importing and exporting data into development environments.

LOGIC AND PROCESS AUTOMATION

* + Describe how to programmatically access and utilize the object schema.
  + Describe the capabilities and use cases for formula fields.
  + Describe the capabilities and use cases for roll-up summary fields.
  + Describe the capabilities of the declarative process automation features.
  + Describe when to use declarative automation features vs. Apex classes and triggers.
  + Describe how to declare variables and constants in Apex and how to assign values using expressions.
  + Describe the primitive and complex Apex data types and when to use them.
  + Describe how to use and apply Apex control flow statement.
  + Describe how to write and when to use Apex classes and interfaces.
  + Describe how to use basic SOSL, SOQL, and DML statements when working with objects in Apex.
  + Describe the basic patterns used in triggers and classes to process data efficiently.
  + Describe when to use and how to write triggers.
  + Describe the implications of governor limits on Apex transactions.
  + Describe the relationship between Apex transactions, the save order of execution, and the potential for recursion and/or cascading.
  + Describe how to implement exception handling in Apex.
  + Describe how to write Visualforce controllers.
  + Describe when and how to use standard Visualforce controllers vs. Apex custom controllers and controller extensions.
  + Describe the programmatic techniques to prevent security vulnerabilities in Apex and Visualforce.
  + Describe how Apex impacts the ability to make declarative changes.

USER INTERFACE

* + Describe how to display Salesforce data using a Visualforce page.
  + Describe the types of web content that can be incorporated into Visualforce pages.
  + Describe how to incorporate Visualforce pages into Lightning Platform applications.
  + Describe the benefits of the Lightning Component framework.
  + Describe the resources that can be contained in a Lightning Component.

TESTING

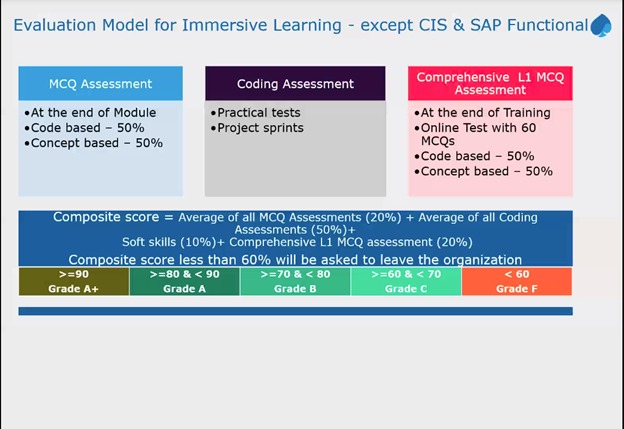
* + Describe the testing framework and requirements for deployment.
  + Describe how to write unit tests for triggers, controllers, and classes.
  + Describe when and how to use various sources of test data.
  + Describe how to execute one or multiple test classes.
  + Describe the differences between invoking Apex in execute anonymous vs. unit tests.

DEBUG AND DEPLOYMENT TOOLS

* + Describe how to monitor and access various types of debug logs.
  + Describe the capabilities and security implications of the Developer Console, Workbench, and Force.com IDE.
  + Describe the different processes for deploying metadata and business data.
  + Describe how the different environments are used in the development and deployment process

**Assessments Marks:**

* **Module 1 (MTT) - 26/40**
* **Module 1 (MPT) - 68/70**
* **Module 2 (MTT) - 26/40**
* **Sprint 1 Project Evaluation - 80/100**
* **Module 3 (MTT) - 33/40**
* **Sprint 2 Project Evaluation – 64/100**
* **Total MCQs Score - 71/100**
* **Total Coding Score - 79/100**
* **Soft Skills Score – 71/100**



**Figure 43: Evaluation Scheme**

**Total Marks: 20% of MCQ Assessment + 50% of Coding Assessment + 10% Soft Skills + 20 % L1 Test**

**Total Marks= 14.2 + 39.5 + 7.1 = 60.8/80**

# Certificate

