

Payment processing Application

## A PROJECT REPORT

**Submitted By**

## Aman Agarwal

**(1900290149012)**

## Submitted in partial fulfillment of the Requirements for the Degree of

MASTER OF COMPUTER APPLICATIONS

## Under the Supervision of

## 

**Submitted to**

**DEPARTMENT OF COMPUTER APPLICATIONS**

## KIET Group of Institutions, Ghaziabad Uttar Pradesh-201206

**(JUNE 2021)**

# DECLARATION

I hereby declare that the work presented in this report entitled “Payment processing Application ", was carried out by me. I have not submitted the matter embodied in this report for the award of any other degree or diploma of other University or Institute.

I have given due credit to the original authors/sources for all the words, ideas, diagrams, graphics, computer programs, experiments, results, that are not my original contribution. I have used quotation marks to identify verbatim sentences and given credit to the original authors/sources.

I affirm that no portion of my work is plagiarized, and the experiments and results reported in the report are not manipulated. In the event of a complaint of plagiarism and the manipulation of the experiments and results, I shall be fully responsible and answerable.

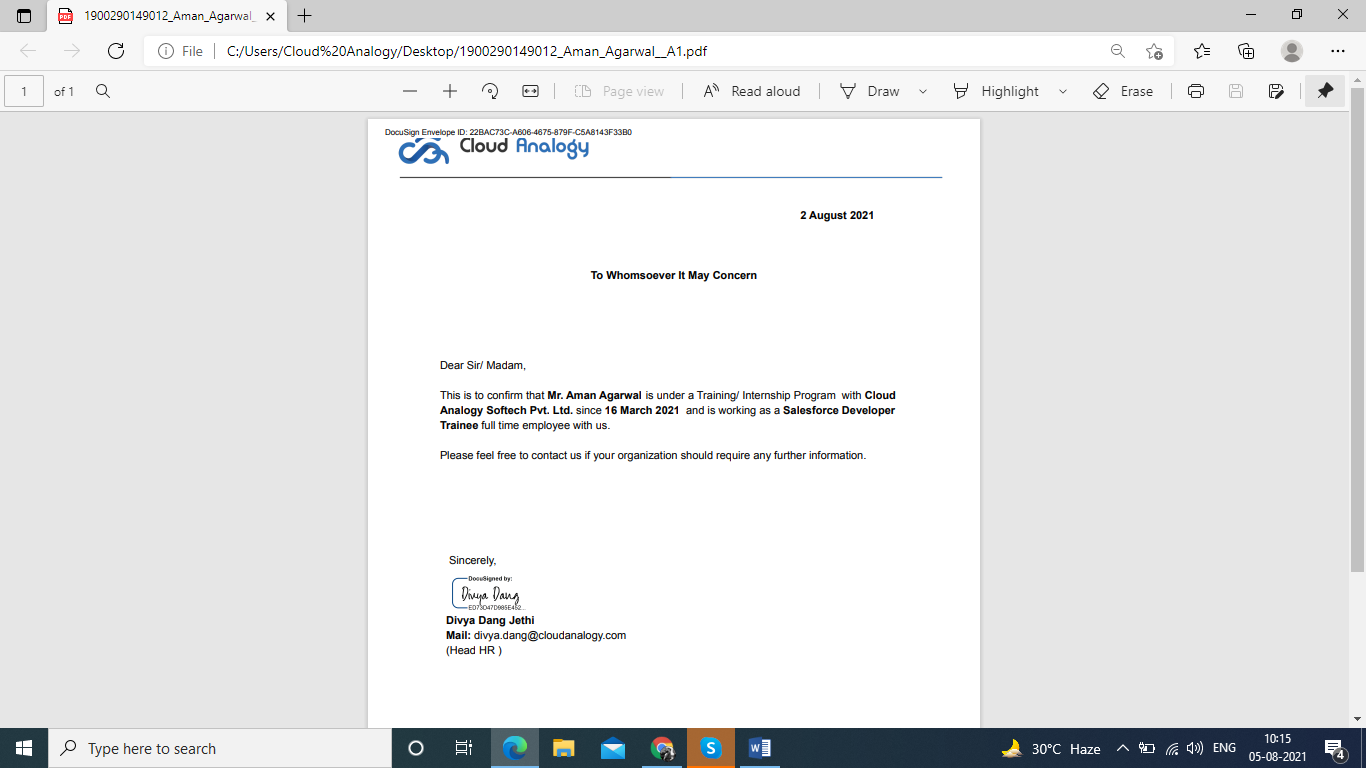
Name : Aman Agarwal

Enroll. No. : 1900290149012

Branch : Master of Computer Applications

**(Candidate Signature)**

**COMPANY CERTIFICATE**



**CERTIFICATE**

Certified that **Aman Agarwal** (**1900290149012**) has carried out the project work presented in this report entitled “**Payment processing Application**” for the award of **Master of Computer Application** from Dr. A.P.J. Abdul Kalam Technical University, Lucknow under my supervision. The report embodies result of original work, and studies are carried out by the student himself and the contents of the report do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University.

**Dr. Sangeeta Arora External Examiner**

Associate Professor

Dept. of Computer Applications

KIET Group of Institutions, Ghaziabad

**Dr. Ajay Kumar Srivastava**

Professor & Head

Department of Computer Applications

KIET Group of Institutions, Ghaziabad

Date:

**Payment processing Application**

**AMAN AGARWAL**

**ABSTRACT**

**“Payment Processing Appication”** is a Salesforce application which is based on cloud computing.

These days we are using databases in which coding is required to enter the data. But this time we have an application which can do our work more simpler and easier i.e Salesforce. This application is used to store a huge amount of data properly and consistently.

The objective of this application is to show that how a normal person who doesn’t even know programming can use this application easily, it is flexible like data can be deleted enter or updated easily.A software project means a lot of experience. I learned a lot through this project. This project has sharpened our concept cloud computing.

This concept of cloud computing has now become a great role to play in today's technical world. These technologies will definitely take database systems far away.

Through this project I learnt so many things can be manage through this application like sales, marketing, commercials and many more things. The only drawback of Salesforce is that it is expensive but to do great work we have to use good technology as today data security is the best and the most essential thing and Salesforce contains that all. things which I also learnt make Data Flow Diagram, State Event Diagram and Activity Diagram with help of our project Mentors.

**ACKNOWLEDGEMENT**

Success in life is never attained single handedly. My deepest gratitude goes to my Project supervisor, **Dr. Ajay Kumar Shrivastava, Professor and Head, Department of Computer Applications** for his guidance, help and encouragement throughout my research work. Their enlightening ideas, comments, and suggestions. Words are not enough to express my gratitude for his insightful comments and administrative help at various occasions.

Fortunately, I have many understanding friends, who have helped me a lot on many critical conditions.

Finally, my sincere thanks go to my family members and all those who have directly and indirectly provided me moral support and other kind of help. Without their support, completion of this work would not have been possible in time. They keep my life filled with enjoyment and happiness.

Aman Agarwal

**1900290149012**

**Table of Content**

Company Certificate ii

Certificate iii

Abstract iv

Acknowledge v

Table of content vi

**CHAPTER 1: Introduction : 5-8**

1.1 Project details 1.2Purpose of the Project

**CHAPTER 2: Technical Feasibility 9-18**

* 1. Cloud computing
  2. Salesforce
  3. Standard Objects

**CHAPTER 3: Creating Objects,Validations,Triggers,API 19-23**

* 1. Creating Salesforce Custom Objects
  2. Creating a Validation Rule

3.3 Apex Triggers  
 3.4 API and web Services  
  
  
 **CHAPTER 4: Design & Code 24-47** 4.1 Source Code   
 4.2 Screenshots   
  
 **CHAPTER 5: Testing 48-49  
  
 CHAPTER 6: Conclusion 50**

**CHAPTER 7: Biblography 51**

**References : 52-53**

## LIST OF FIGURES

2.1 Benefits of Cloud computing 10

2.2 Salesforce 13

2.3 Standard Objects 14

3.1 Custom Object 19

3.2 Validation rules 20

3.3 Apex Triggers 21

3.4 API and Web Services 22

4.1 Lightning Components. 24

4.2 Standard forms and Validations. 39

## CHAPTER 1

**INTRODUCTION**

## Project Details:

“**Institute Application**” is taking a concept from “**Cloud Computing**” and Salesforce was founded by **“Marc Benioff”**.

We integrate Stripe Payment Gateway with Salesforce and using Salesforce we can create customers, products, payment methods, subscriptions and payments for Stripe, in fact Stripe Payments is a payment processing platform. It allows you to transfer money from a customer's bank account to your business account via credit or debit card transactions and requires both a payment gateway and a payment processor for you to process the transaction online. The gateway securely transmits the customer's credit card payment information to the processor, which actually processes the transaction. The money is temporarily transferred from the customer's bank to the merchant account (in this case a third party processor like Stripe, it's a consolidated account - more on that later), where credit card-related charges are deducted from the amount is. The payment is then sent to the merchant's bank account.

## Purpose:

Salesforce provides you with the fastest path from Idea to App. You can concentrate on building your app using Salesforce tools, rather than building the infrastructure and tools yourself. This can save you years of time and millions of dollars.

Salesforce customers generally say that it’s unique for three major reasons:

**Fast** – Traditional CRM software can take more than a year to deploy, compare that to months or even weeks with Salesforce.

**Easy** – Salesforce wins in the easy to use category hands down. You can spend more time putting it to use and less time figuring it out.

**Effective** – Because it is easy to use and can be customized to meet business needs, customers find Salesforce very effective.

Salesforce is in the cloud, so your team can use it from anywhere with access to the internet. If you are a business that is rapidly changing or you are a seasoned company that’s been around for years, your business is probably changing too. Salesforce is completely scalable to your growth.

Salesforce seamlessly integrates with 3rd party apps. If you want to integrate Salesforce with Gmail you can do it, if you want to integrate it with your accounting software you can do that too. On the other hand, integration is tough with other CRMs. Salesforce is affordable, especially if you consider its vast variety of capabilities. Even startups and small businesses can use Salesforce.

## Scope:

[The Customer Success Platform To Grow Your Business](http://salesforce.com/) is a cloud-based CRM system that allows salespeople to track their sales, support people to track their cases, and the entire company's employees to collaborate with each other. [Salesforce.com: The Customer Success Platform To Grow Your Business](http://salesforce.com/) is also a platform on which one can build new applications for purposes other than CRM.

In these terms, salerforce has a lot of of scope in the coming years. Salesforce is that sector which increases the sale of the company ‘in real terms’. There is no reason to behold this sector. No one can underestimate or oversee its programming. Every IT or any other company has to invest in the working of it. Hence, if to be joined or taking any course on it. Proceed without a second thought. Every other sector can be taken over by AI, but salesforce is that sector which needs real human brain.

## CHAPTER 2

**CLOUD COMPUTING**

## About:

Cloud computing relies on software for distributed batch and stream processing, as well as distributed storage. This chapter focuses on an oft*-*ignored angle of assuredness: performance assuredness. A significant pain point today is the inability to support reconfiguration operations, such as changing of the shard key in a sharded storage/database system, or scaling up (or down) of the number of virtual machines (VMs) being used in a stream or batch processing system. We discuss new techniques to support such reconfiguration operations in an online manner, whereby the system does not need to be shut down and the user/client*-*perceived behavior is indistinguishable regardless of whether a reconfiguration is occurring in the background, that is, the performance continues to be assured in spite of ongoing background reconfiguration. Next, we describe how to scale*-*out and scale*-*in (increase or decrease) the number of machines/VMs in cloud computing frameworks like distributed stream processing and distributed graph processing systems, again while offering assured performance to the customer in spite of the reconfigurations occurring in the background. The ultimate performance assuredness is the ability to support SLAs/SLOs (service*-*level agreements/objectives) such as deadlines. We present a new real*-*time scheduler that supports priorities and hard deadlines for Hadoop jobs. We implemented our reconfiguration systems as patches to several popular and open*-*source cloud computing systems, including MongoDB and Cassandra (storage), Storm (stream processing), LFGraph (graph processing), and Hadoop (batch processing).

Cloud computing has become the industry standard for rapid application deployment, scalable server support, mobile and distributed services, and it provides access to (theoretically) infinite resources. Unfortunately, researchers are still trying to converge towards cross-provider cloud computing frameworks to enable compatibility and seamless resource transition between cloud providers. Moreover, users are restricted to using the provider-specific pre-configured options of resources and services, irrespective of their current needs. At the same time, cloud services are provided as a direct service from the providers to the clients. This creates a segregated cloud market clientele, and non-negotiable pricing

Strategies for the cloud services. In this paper, we propose Jugo, a generic architecture for cloud composition and negotiated service delivery for cloud users. Jugo acts as a match-maker for service specifications from the users with the currently available assets from the cloud providers. The engagement of a middle-man as an opaque cloud service provider will create a better opportunity for cloud users to find cheaper deals, price-matching, and flexible resource specifications, with increased revenue and higher resource utilization for the cloud service providers.

Many enterprises in industries start using Cloud Computing for their IT infrastructure services. This adoption of Cloud Computing is a part of the enterprise transformation which is the migration from a legacy IT environment to Cloud Computing. On the other hand, one of major targets is an industry solution which provides a critical business service to their end customers. This paper proposes Industry Cloud which is the enhanced design of Cloud Computing for industry solutions. It efficiently supports industry solutions for enterprise business requirements. The paper describes Industry Cloud with a requirement analysis of industry solutions, those adopted functions, and three use case scenarios in the electronics and retail industry. The contribution of the paper is the analysis of industry wide requirements, the definition of Industry Cloud with a common function among industry solutions and the usage with use case scenarios.[3]

The complexity of Cloud infrastructures is increasing every year, requiring new concepts and tools to face off topics such as process configuration and reconfiguration, automatic scaling, elastic computing and healthiness control. This paper presents a Smart Cloud solution based on a Knowledge Base, KB, with the aim of modeling cloud resources, Service Level Agreements and their evolution, while enabling the reasoning on cloud structures and implementing strategies of efficient smart cloud management and intelligence. The solution proposed is composed of Smart Cloud Engine, SCE, the Knowledge Base, KB, and the Supervisor and Monitoring module for data acquisition. It can be easily integrated with any cloud configuration manager, cloud orchestra or, and monitoring tool, since the connections with these tools are performed by using REST calls and XML files.

Current Cloud Computing is primarily based on proprietary data centers, where hundreds of thousands of dedicated servers are setup to host the cloud services. In addition to the huge number of dedicated servers deployed in data centers, there are billions of underutilized Personal Computers (PCs), usually used only for a few hours per day, owned by individuals and organizations worldwide. The vast untapped compute and storage capacities of the underutilized PCs can be consolidated as alternative cloud fabrics to provision broad cloud services, primarily infrastructure as a service. This approach, thus referred to as "no data center" approach, complements the data center based cloud provision model. In this paper, we present our opportunistic Cloud Computing system, called cuCloud, that runs on scavenged resources of underutilized PCs within an organization/community. Our system demonstrates that the "no data center" solution indeed works. Besides proving our concept, model, and philosophy, our experimental results are highly encouraging.[5]

Whatever one public cloud, private cloud or a mixed cloud, the users lack of effective security quantifiable evaluation methods to grasp the security situation of its own information infrastructure on the whole. This paper provides a quantifiable security evaluation system for different clouds that can be accessed by consistent API. The evaluation system includes security scanning engine, security recovery engine, security quantifiable evaluation model, visual display module and etc. The security evaluation model composes of a set of evaluation elements corresponding different fields, such as computing, storage, network, maintenance, application security and etc. Each element is assigned a three tuple on vulnerabilities, score and repair method. The system adopts “One vote vetoed” mechanism for one field to count its score and adds up the summary as the total score, and to create one security view.

We implement the quantifiable evaluation for different cloud users based on our G-Cloud platform. It shows the dynamic security scanning score for one or multiple clouds with visual graphs and guided users to modify configuration, improve operation and repair vulnerabilities, so as to improve the security of their cloud resources.

To move applications to the cloud is not only a technical decision but also a business-oriented decision, in which both business and technical factors (e.g. transformation effort,

multi-tenancy and auto-scaling enablement, scalability and extensibility) should be considered. However, existing approaches and tools do not support a consumable business oriented cloud transformation decision to select more suitable transformation solution with the right cloud delivery model, services type, affordable transformation effort and etc. In this paper, we introduce a practical three-step approach and a tool, CTA (Cloud Transformation Advisor) to enable decision makers to identify the most suitable cloud transformation solution to satisfy their business goals based on a well-structured cloud transformation knowledge

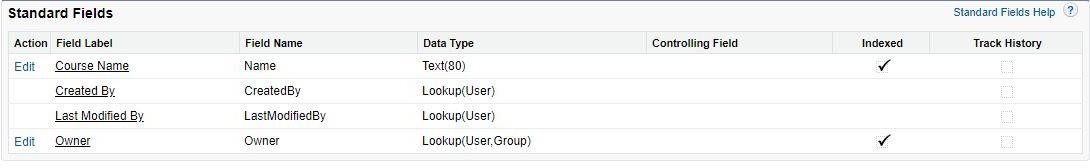
**2.2 SALESFORCE**

The Salesforce Platform stores data in relational tables. The records in these tables contain data for the structure of the platform itself as well as user created data. For example, the data about the configuration and settings of an account are already in-built as a relational table. But you can also create your own tables to store data specific to your business like the 'dispatch schedule' for a week assuming you are a courier company.

These relational tables are roughly referred to as API Objects or only objects in Salesforce.

There are three kinds of Salesforce objects.

* Standard Objects − The objects already created for you by the Salesforce platform.
* Custom Objects − These are the objects created by you based on your business processes.
* External Objects − The objects which you create map to the data stored outside your organization.
  1. **Standard Objects**



These are the objects which already exist in the Salesforce platform to manage the configurations and settings of the environment. Once you log in to the salesforce platform, you can see the available objects.

**Important Standard Objects**

In this section, we will discuss the important standard objects in Salesforce. The following table lists down the objects −

|  |  |  |
| --- | --- | --- |
| **Object Name** | **Meaning** | **Usage** |
| Account | Represents an individual account, which is an organization or person involved in the business like customers,competitors,partners, etc. | Use this object to query and manage accounts in your organization. |
| Account History | Represents the history of changes to the values in the fields of an account. | Use this object to identify changesto an account. |
| Case | Represents a case, which is a customer issue or problem. | Use the case object to manage casesfor your organization. |
| Contact | Represents a contact, which is an individual associated with anaccount. | This object is used to manage individuals who are associated with an Account in the organization. |

|  |  |  |
| --- | --- | --- |
| ser | Represents a user in the organization. | This object is used to query information about users and also helps to provide and modify the information concerning the users. |
| Asset | Represents an item of commercial value, such as a product sold by the company or a competitor that a customer has purchased and installed. | This object is used to track assets previously sold into customer accounts. With asset tracking, a client application can quickly determine which products were previously sold or are currently installed at a specific account. |
| Domain | Read-only object that represents a custom Web address assigned to a site in your organization. | This read-only object is used to object to query the domains that are associated with each website in your organization. |

As IT technology advanced, a new style of innovation emerged, in which a leading innovation company invites end-users to its open software service platform. With respect to this type of innovation, a lot of innovation studies were performed to understand the structure of the interaction among users and the platform provider from the perspective of network science. By concentrating only on the internal mechanisms among agents, the previous studies miss to consider innovation through collective intelligence. A platform provider plays an important role in the innovation. In this research, we investigate the structure of a service network with empirical data gathered from Salesforce.com App Exchange and discuss the role of a platform provider in innovation through collective intelligence. Our results suggest that the platform provider led the innovation in the initial period and, then, third party developers became gradually innovation leaders. Our findings are expected to re-orient the research focus from internal mechanisms to the role of platform providers.[8]

With the appearance of distributed computing, associations are hoping to move their Customer Relationship Management (CRM) applications from an On-Premise environment or we can say local servers to an On-Demand environment that is on cloud server. On-Premise environment is when association has the framework and programming inside their system. In On-Demand environment, an outsider has the base and programming and charges the relationship in light of its participation. Salesforce is the principle On-Demand CRM thing.

The advantages of cloud while supporting real-time service systems using the Salesforce platform. We build here a service management platform for the Polish Billiards and Snooker Association (PBSA), based on a real-time system located in a cloud. It allows PBSA managers to accomplish tasks in this system on-demand. And, it is deployed as a private cloud to grant an access only to the employees from the snooker organization.

In a recent scenario, IT industries are growing with the help of proper Utilization of available resources. The IT giants like Microsoft, Infosys, IBM, Oracle, & TCS are switching from theirs on premises IT setups to the cloud. Cloud computing is replacing the traditional model in which software applications installed on on-premise hardware, from desktop computers to server rooms, depending on the size of the business. The proposed work is about the cloud platform which is going to change all the traditional views of software, application, and product development Technologies. Salesforce.com is one of the best cloud providers available in the recent scenario. There are number of reasons why IT industries are switching to the Cloud. And there are number of reasons why Industries have to think to adopt Salesforce.com cloud. The proposed work is about to focus on important and common features of salsforce.com. These features are common for any developer to learn and use in to software, application and product development in salesforce.com. The goal of this proposed work is to show the available resources in the salesforce.com which are still new for the developers. This an approach to make people familiar with the salesforce.com cloud provider.

Summary form only given as follows. The strongest predictors of new product success is understanding market requirements early in the new product development (NPD) process. A direct salesforce

is one of the best sources of new product ideas and market information, although not fully leveraged in many firms. A recent study of 248 salespeople in nineteen high-tech firms indicates wide variation in NPD involvement across and within a firm's salesforce. The study revealed that at the organization level, the length of the NPD cycle was associated with salesforce-initiated and NPD-headquarters initiated activities. In particular, the longer a firm's NPD cycle for product improvements, the lower the involvement the salesforce has in headquarters initiated involvement activities. A significant relationship exists between a firm's new product cycle time and the level of involvement in salesforce-initiated NPD predevelopment activities. At the salesperson level, several factors affect involvement in NPD activities. Most significant was the relationship between salesperson involvement and the distance between a salesperson's location and the NPD office site. The study shows that organizations can affect the degree of involvement that their salesforce or individual salespeople have in early phases of NPD.

Motivation is important because it determines and individual's effort toward performing a task and it leads to an optimistic and challenging attitude at work place. Incentive motivation is concerned with the way goals influence behavior. Incentive is the promise or stimulus for greater action. Incentives are something that is given in addition to wages. It means additional remuneration or benefit to an employee in recognition of achievement or better work. Organizations in emerging markets are more likely to give group rewards and generally do not encourage risk taking as much as developed economies. Money motivates employees to some extent but there are other powerful sources of motivation such as interesting and challenging tasks and assignments, praise, good interpersonal relations etc. This paper analyzes incentives plan structure of sales people in Croatia. Advantages and disadvantages of different compensation plans and benefits for salespeople are discussed.

Modern business decision models are often very complicated due to a deluge of information. Evaluation and diagnostics of such decision models is extremely challenging due to many factors, including the complexity and volume of data. In addition, since there is no ideal data sample to construct a control group for comparison studies, performance evaluation and diagnostics of business actions can easily be distorted by selection bias. In this paper, we design a framework to analyze this sample bias issue under a practical business scenario. In particular, we focus on: a) identification of the key factors which drive selection bias during the business decision; b) evaluation of the performance of business actions with consideration of the identified selection bias. We evaluate baseline analytics tools on the worldwide sales- force data of a large global corporation and clearly demonstrate that the selection bias issue makes the

usual evaluation very unstable and not trustable. However, by removing such detected sample bias, our framework can generate reasonable diagnostics results across different dimensions. The implemented analysis tool was applied to a worldwide business opportunity dataset of a multinational Fortune 500 corporation; the analytics results clearly show the significance of such a bias detection-based evaluation framework for sales-force optimization.

Estimating generalizable relationships between actions and results from historical samples, especially when there is a level of noise or randomness in that signal, is an important problem to address before making decisions on actions to take. Many business analytics problems require the optimal assignment of limited resources to actions and activities to maximize some result or objective such as profit. We present a novel approach to solving this class of analytics problems by modeling the relationship between resource effort and expected return as a dose-response signal and formulating its causal estimation as one of kernel regression. The estimated expected value and variance of the result are then used to optimize resource allocation so as to maximize expected response while minimizing the risk around response subject to business constraints. We apply this approach to the task of optimally assigning salespeople to enterprise clients using real-world data, and show that profit can be substantially increased with fewer salespeople and reduced risk.

**CHAPTER 3**

**HOW TO CREATE OBJECTS, VALIDATIONS AND TRIGGERS**

* 1. **Creating a Salesforce Custom Object**
* First, follow this path: Setup > Build > Create > Objects > on the screen, click on the new custom object button > enter label name, plural label, and object name > enter record name as a data type.
* To create a record name, first, consider the two data types:
* Text
* Auto-number

You also have some optional features while creating objects:

* Allow reports: If you tick off this checkbox, then only these objects will be available to make reports.
* Allow activities: If you tick off this checkbox, then you are able to make activities on this object.
* Track field history: When you tick off this checkbox, then you can merely track fields. You can track up to 20 fields for a single object.

Now, you have the following deployment modes:

* In development: If you opt for this, then this object will remain in the development mode. It will not be present for deployment.
* Deployed: When you opt for this, the object will be available for deployment.

The following options are available only when you are creating a custom object for the first time:

* Add notes and attachments related list
* Launch a new custom tab wizard after saving this custom object After completing all these settings, press the Save button.

If you do not select ‘Launch new custom tab wizard’ from the object creation page, then the object would be saved without the tab appearance. In such a case, first, you need to make a tab for this object. On the

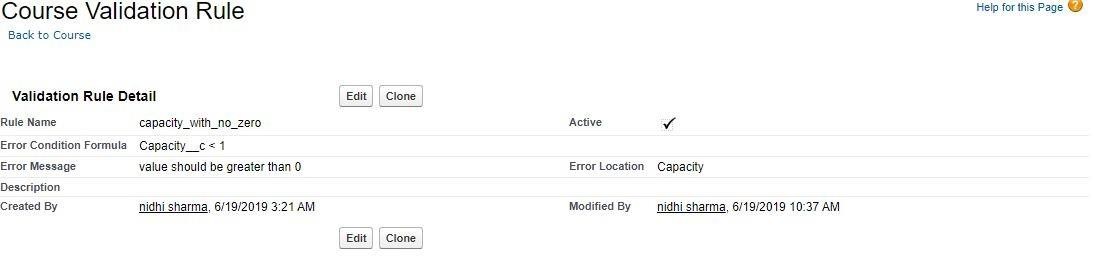
other hand, if you select this option, the object would be saved, along with a tab would be created, which will be visible to you.

* 1. **Creating a Validation Rule**
* From Setup, go to Object Manager and click Account
* In the left sidebar, click Validation Rules.
* Click New.
* Enter the following properties for your validation rule
* Rule Name: Account\_Number\_8\_Characters
* Error Condition Formula:

LEN( AccountNumber) != 8

* Error Message: Account number must be 8 characters long.
* To check your formula for errors, click Check Syntax.
* Click Save to finish.





* 1. **Apex Triggers**

A **trigger is** an **Apex script** that executes before or after data manipulation language (**DML)** events occur. Apex triggers enable you to perform custom actions before or after events to record in Salesforce, such as insertions, updates, or deletions. Just like database systems support triggers, Apex provides trigger support for managing records.  
  
**Here is a list of trigger events in Salesforce:**

* before insert
* before update
* before delete
* after insert
* after update
* after delete
* after undelete

**Here is list of context variables in triggers:**

* **isInsert**: Returns true if this trigger was fired due to an insert operation, from the Salesforce user interface, Apex, or the API.
* **isUpdate**: Returns true if this trigger was fired due to an update operation, from the Salesforce user interface, Apex, or the API.
* **isDelete**: Returns true if this trigger was fired due to a delete operation, from the Salesforce user interface, Apex, or the API.
* **isBefore**: Returns true if this trigger was fired before any record was saved.
* **isAfter**: Returns true if this trigger was fired after all records were saved.
* **isUndelete**: Returns true if this trigger was fired after a record is recovered from the Recycle Bin (that is, after an undelete operation from the Salesforce user interface, Apex, or the API.)
* **new**: Returns a list of the new versions of the sObject records. This sObject list is only available in insert, update, and undelete triggers, and the records can only be modified in before triggers.
* **newMap**: A map of IDs to the new versions of the sObject records. This map is only available in before update, after insert, after update, and after undelete triggers.
* **old** : Returns a list of the old versions of the sObject records. This sObject list is only available in update and delete triggers.
* **oldMap**: A map of IDs to the old versions of the sObject records. This map is only available in update and delete triggers.

## 3.4 API and Web Services.

## ****3.4.1 What is Integration?****

Integration is a process of connecting two or more applications. Enterprise system uses many applications, many or most of which are not designed to work with one another out of the box. Each application can have data, business logic, presentation, and security layers, all of which are possible targets for integration.

We did one session in Apex hours. In [that session](http://www.apexhours.com/integrating-with-salesforce/)we covered the Basic of Integration in Salesforce. Here is agenda which covered in that session

* Web Communication Fundamentals
* Understanding REST Vs SOAP
* Message Exchange Formats – XML & JSON
* Performing Callouts to External Services
* HTTP Request & HTTP Response

## ****3.4.2 What is webservices?****

Web service is a standardized medium to propagate communication between the client and server applications on the World Wide Web. Web services provide a common platform that allows multiple applications built on various programming languages to have the ability to communicate with each other.

Webservices is a functionality or code which helps to us to do integration. Web services are open standard (XML, SOAP, HTTP, etc.) based web applications that interact with other web applications for the purpose of exchanging data.

### **Type of Web Service**

There are mainly two types of web services.

* SOAP web services.
* RESTful web services.

### **3.4.3 What is REST API ?**

* Representational State Transfer is a style of software architecture for distributed hypermedia systems.
* REST API has a lightweight request and response framework
* Simple, easy to use and powerful web service based on RESTful principles.
* Expose functionality via REST resources and HTTP methods.
* (CRUD) records, search or query your data, retrieve object metadata, and access information about limits in your org.
* REST API supports both XML and JSON.
* Mobile and Web appsRest resource is referenced using URI, abstraction of information, access using HTTP methods

### **3.4.4 What is SOAP API ?**

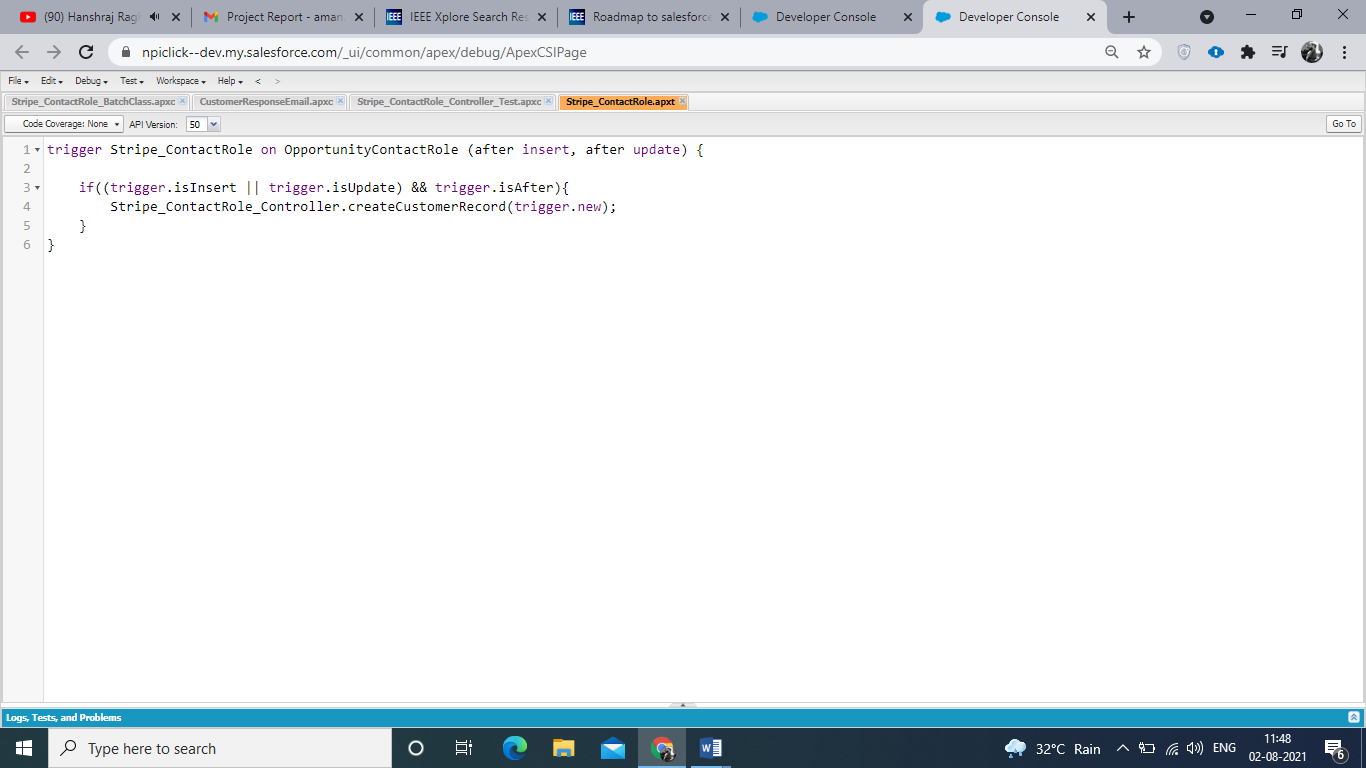
* Application layer protocol used to exchange structured information between systems
* It uses a Web Services Description Language (WSDL) file to rigorously define the parameters for accessing data through the API.
* SOAP API supports XML only.
* Because SOAP API uses the WSDL file as a formal contract between the API and consumer, it’s great for writing server-to-server integrations.
* Access to Salesforce data and business logic
* Handles medium data volumes
* Updates multiple records with single

**CHAPTER 4**

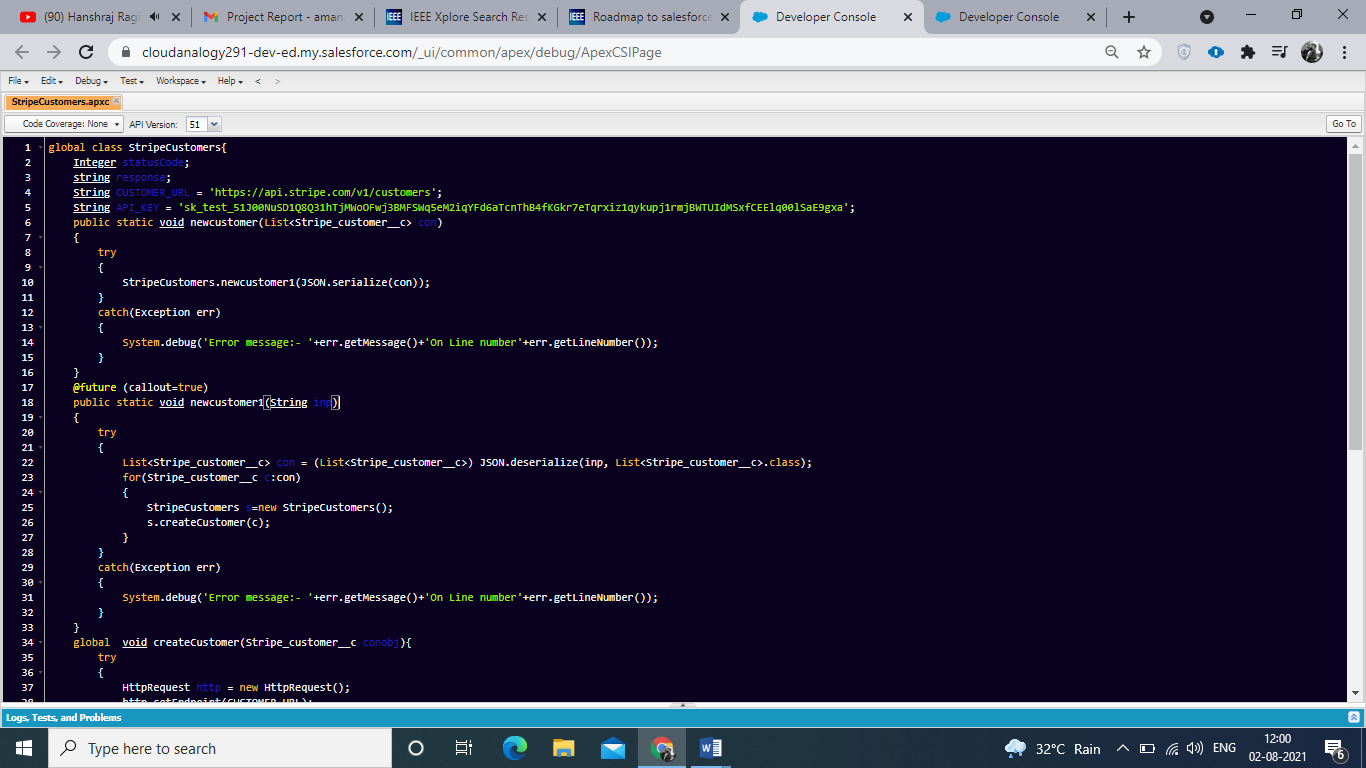
**DESIGN AND CODE**

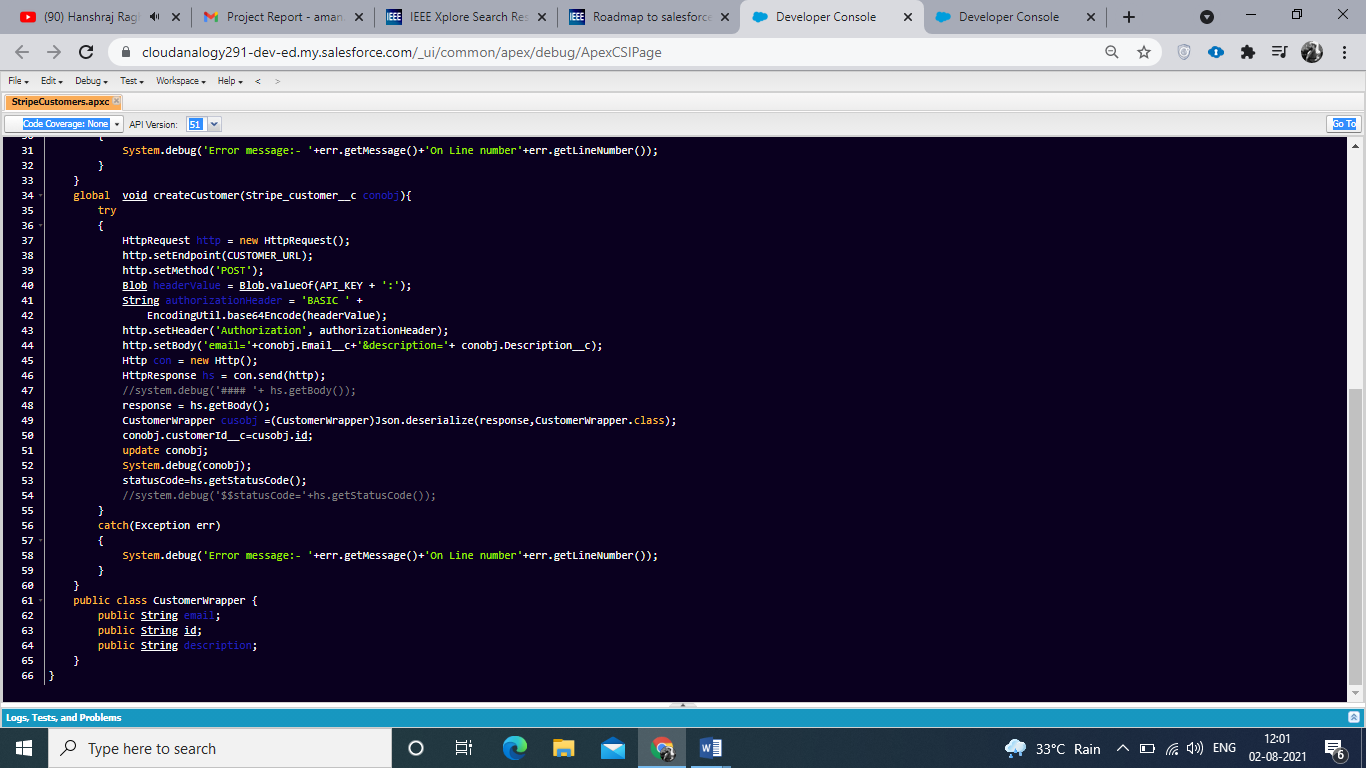
**4.1 Source Code**

**4.1.1 Apex Trigger for create customer.**

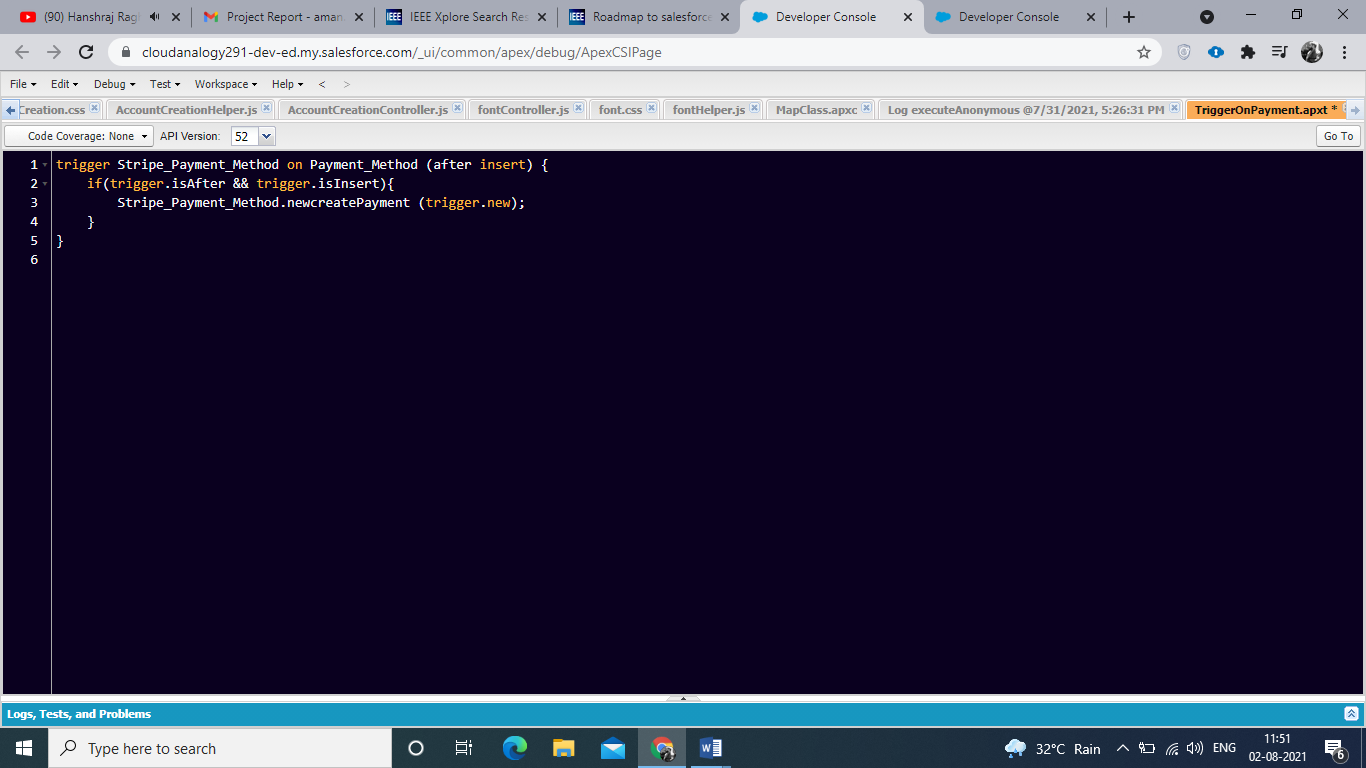


**4.1.2 Apex Class for create customer.**

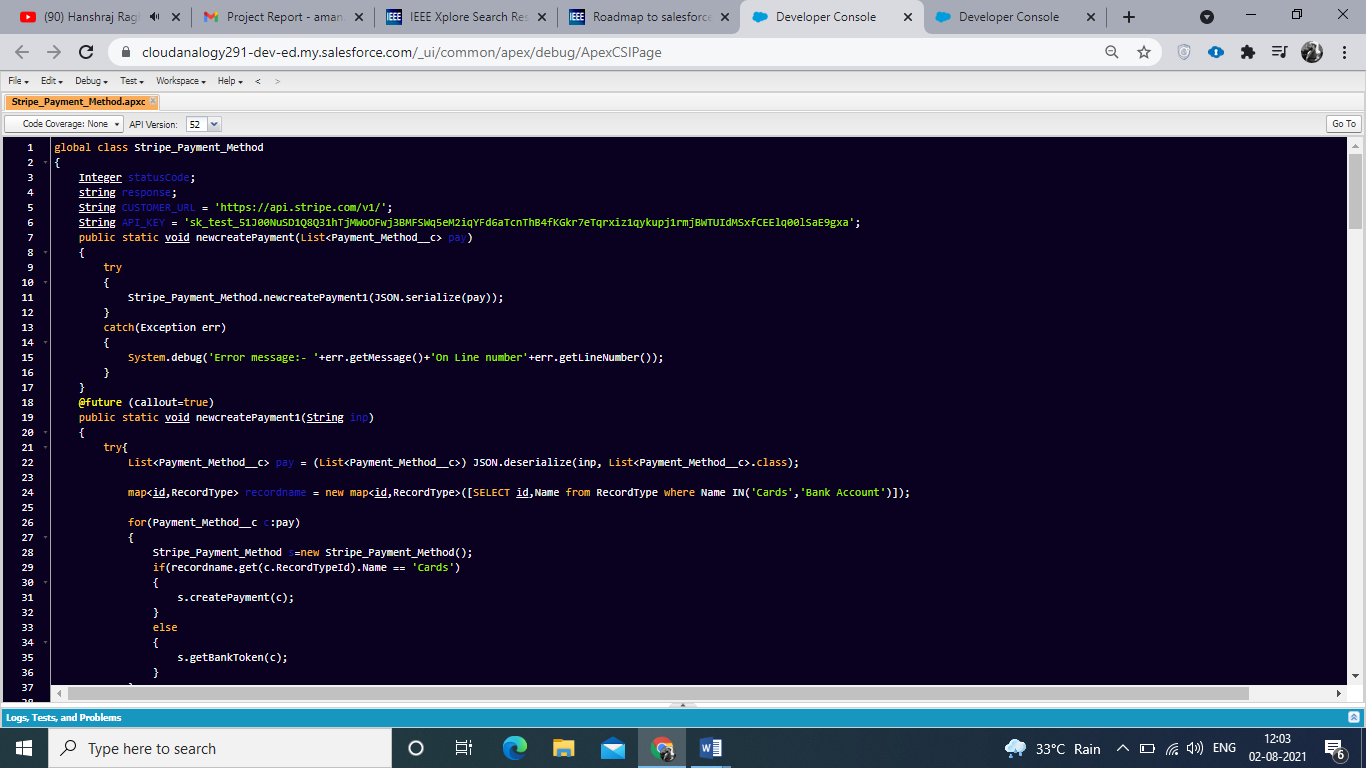


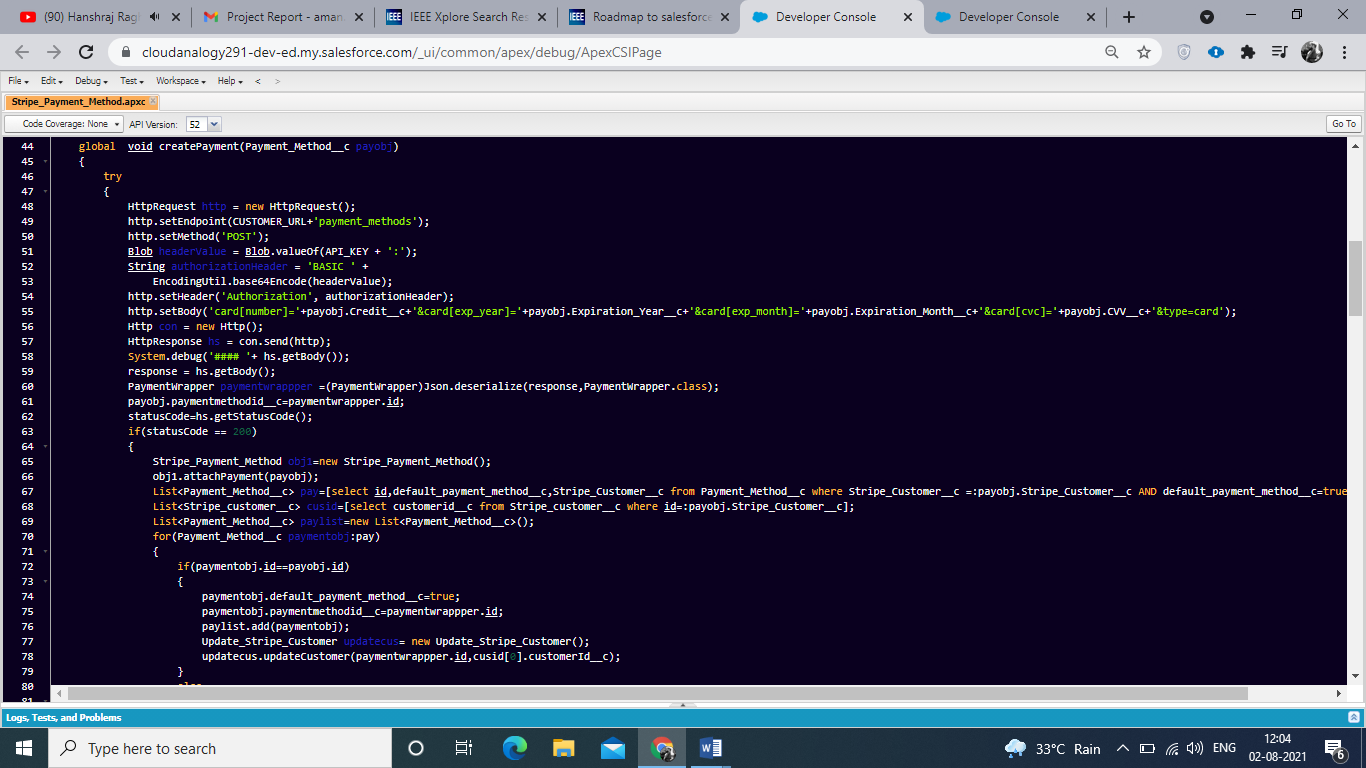


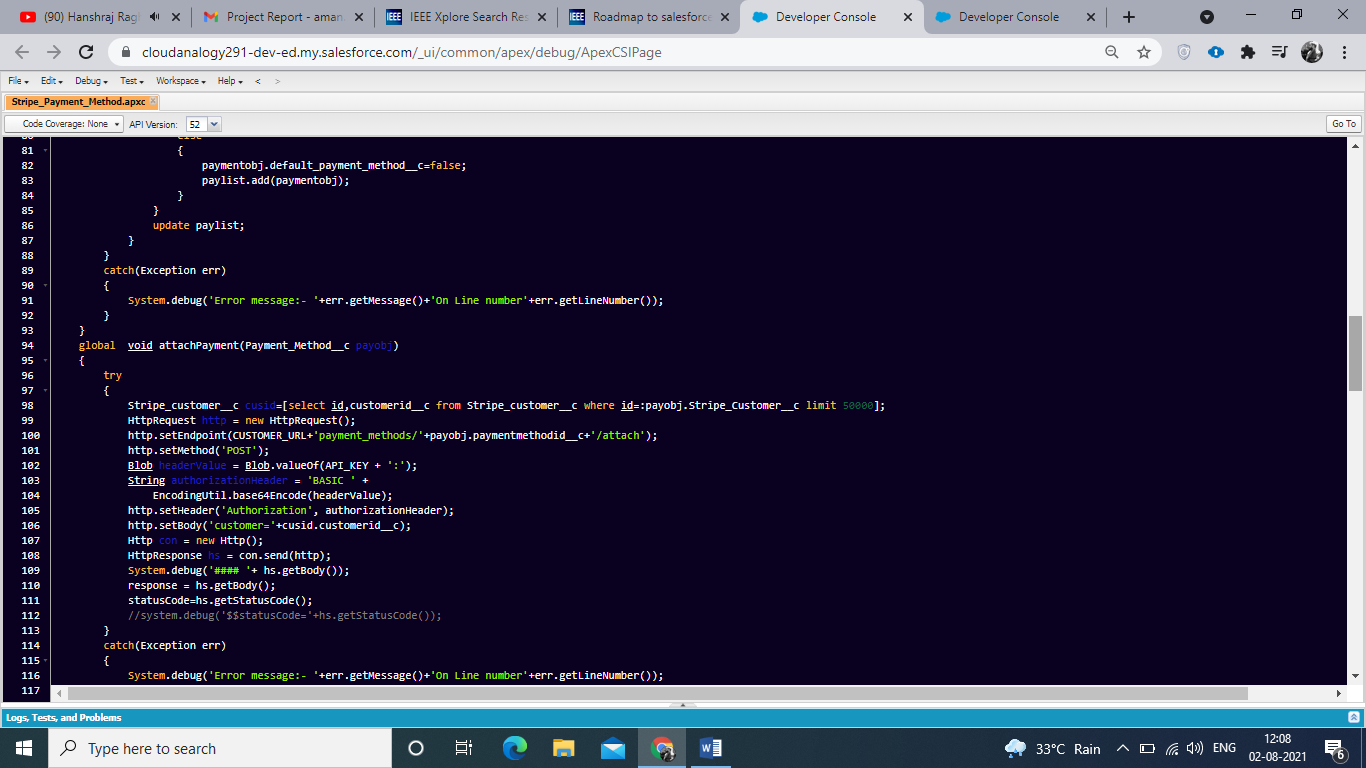
**4.1.3 Apex Trigger for create payment method.**

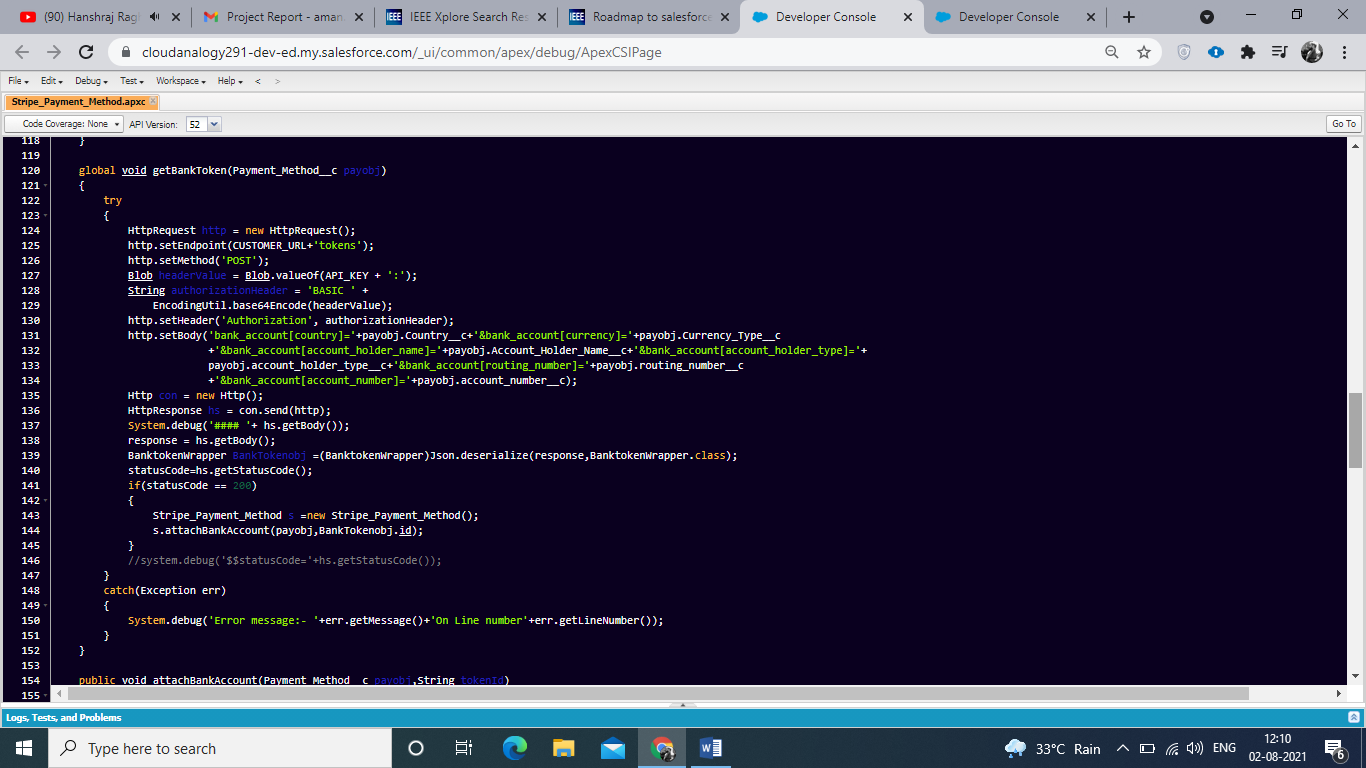


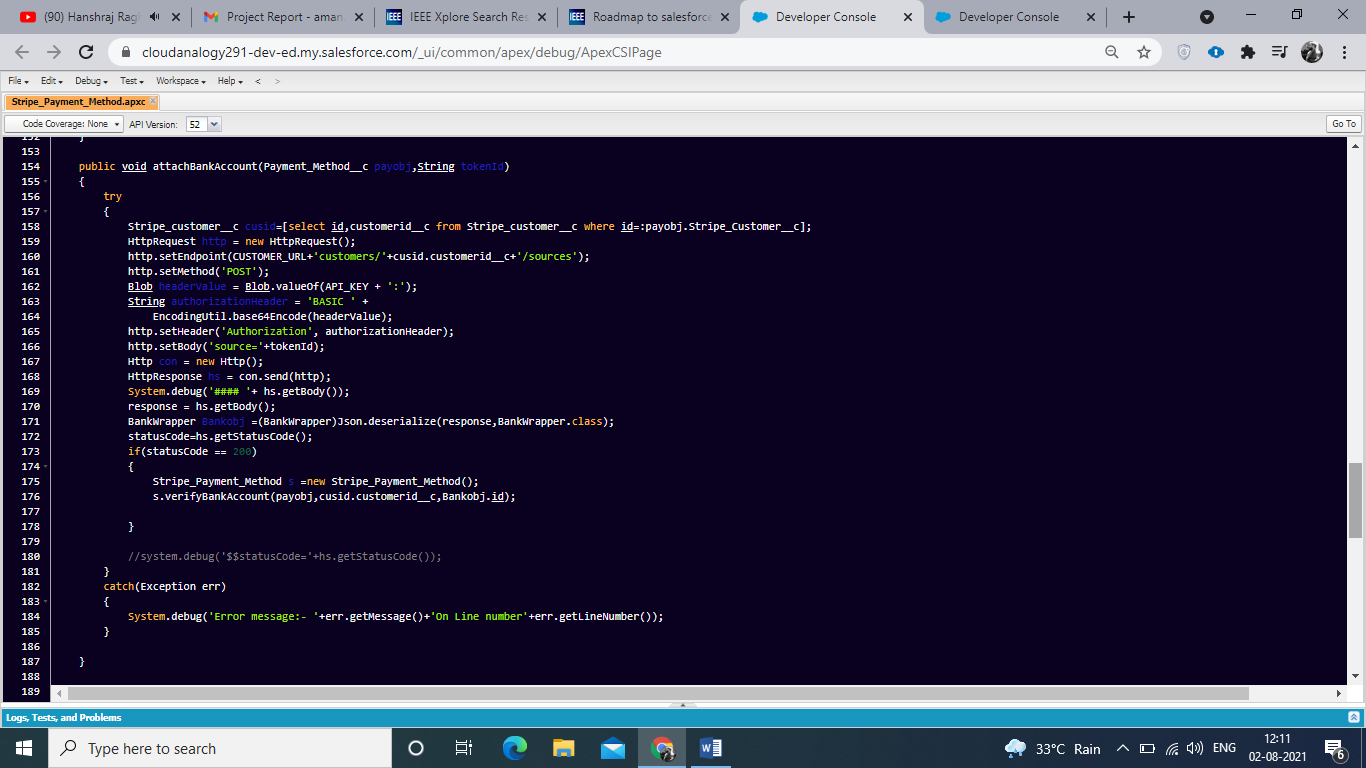
**4.1.4 Apex Class for create payment method.**

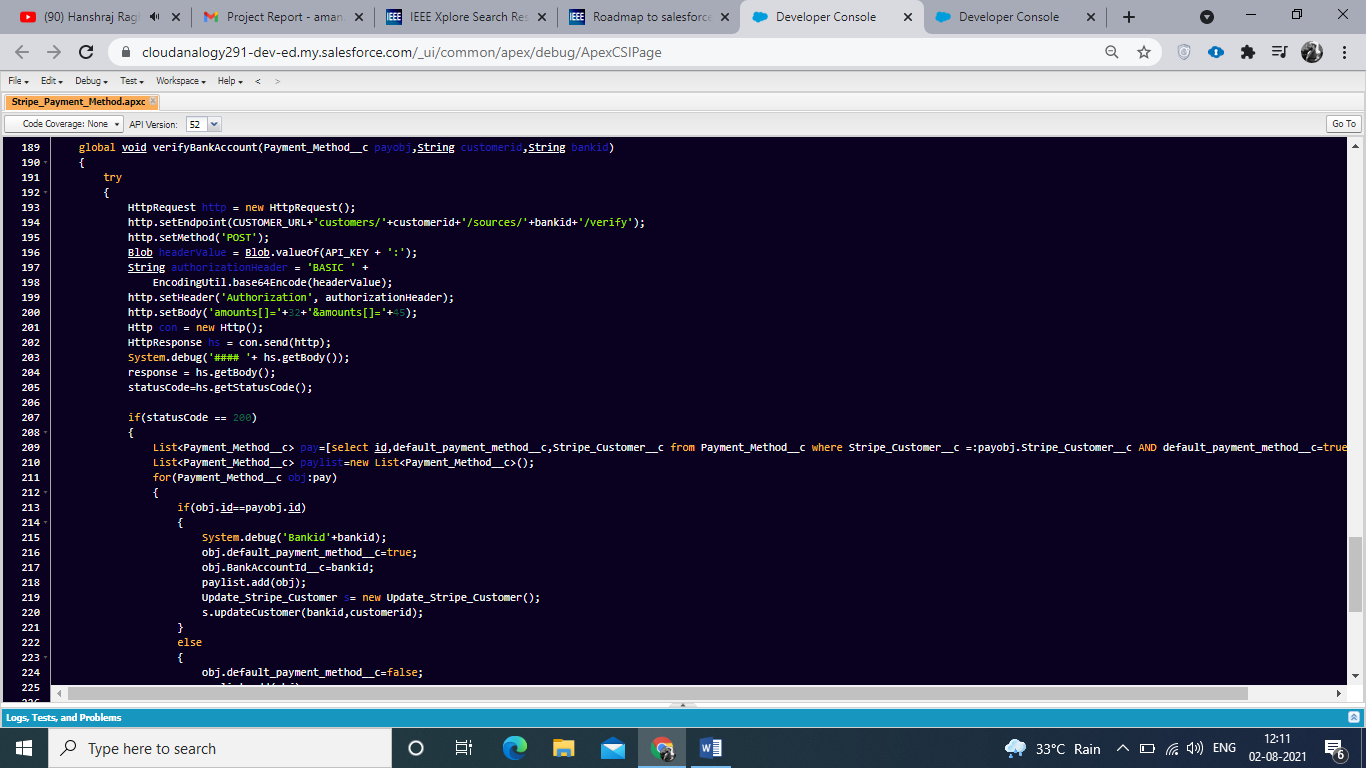


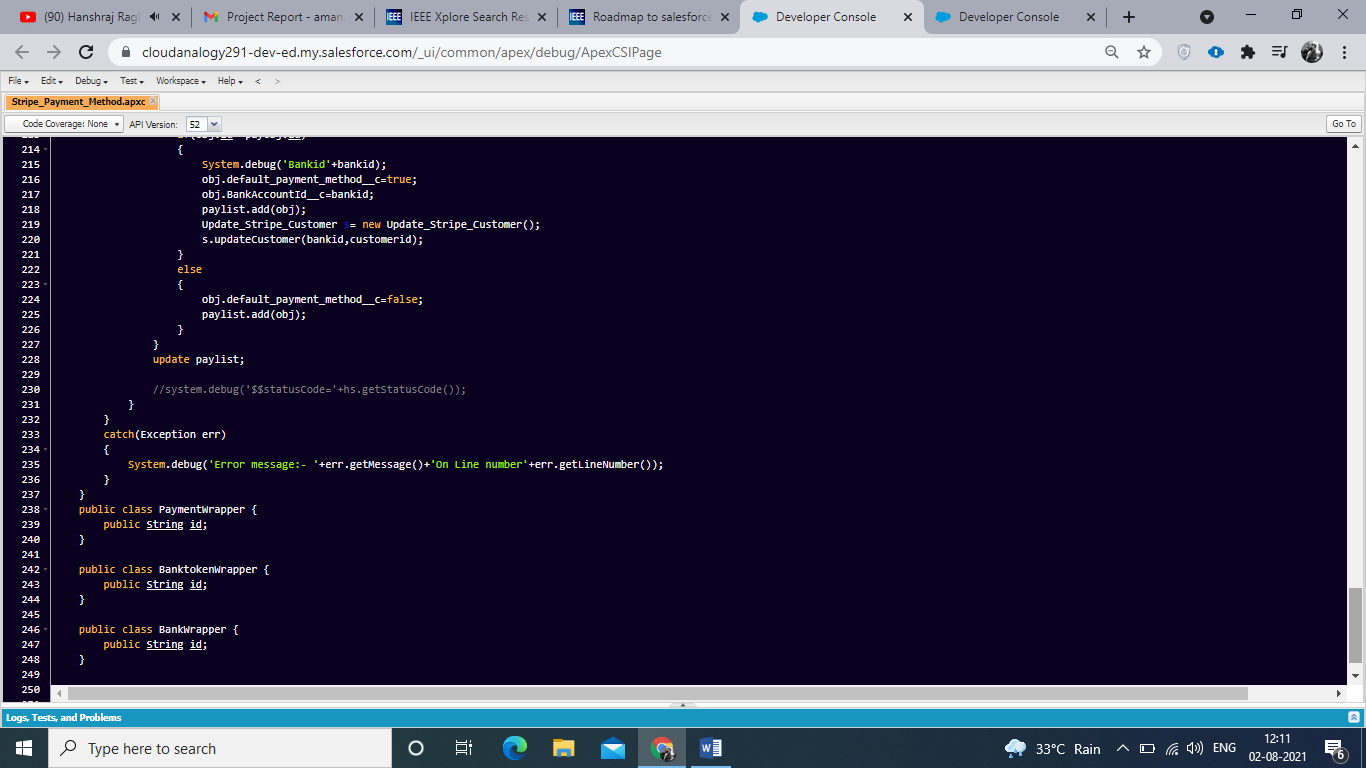




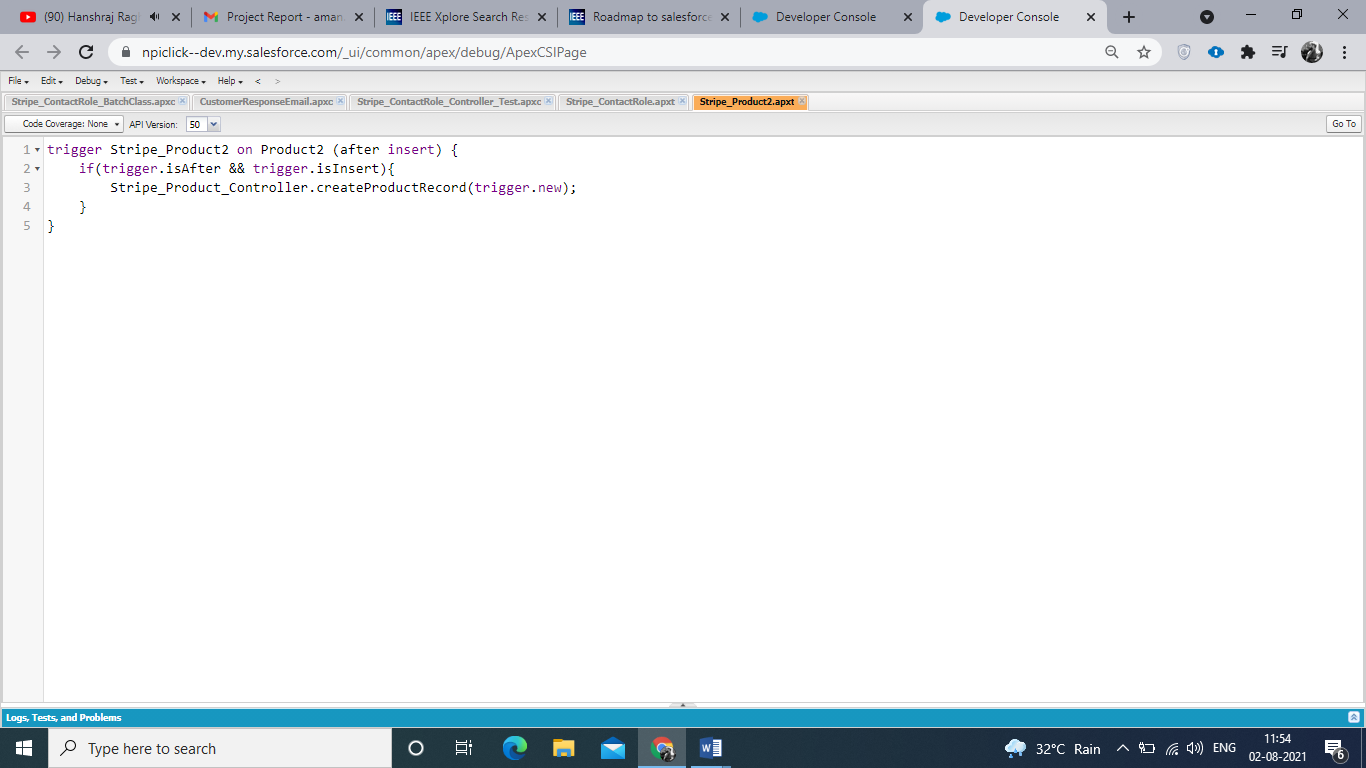




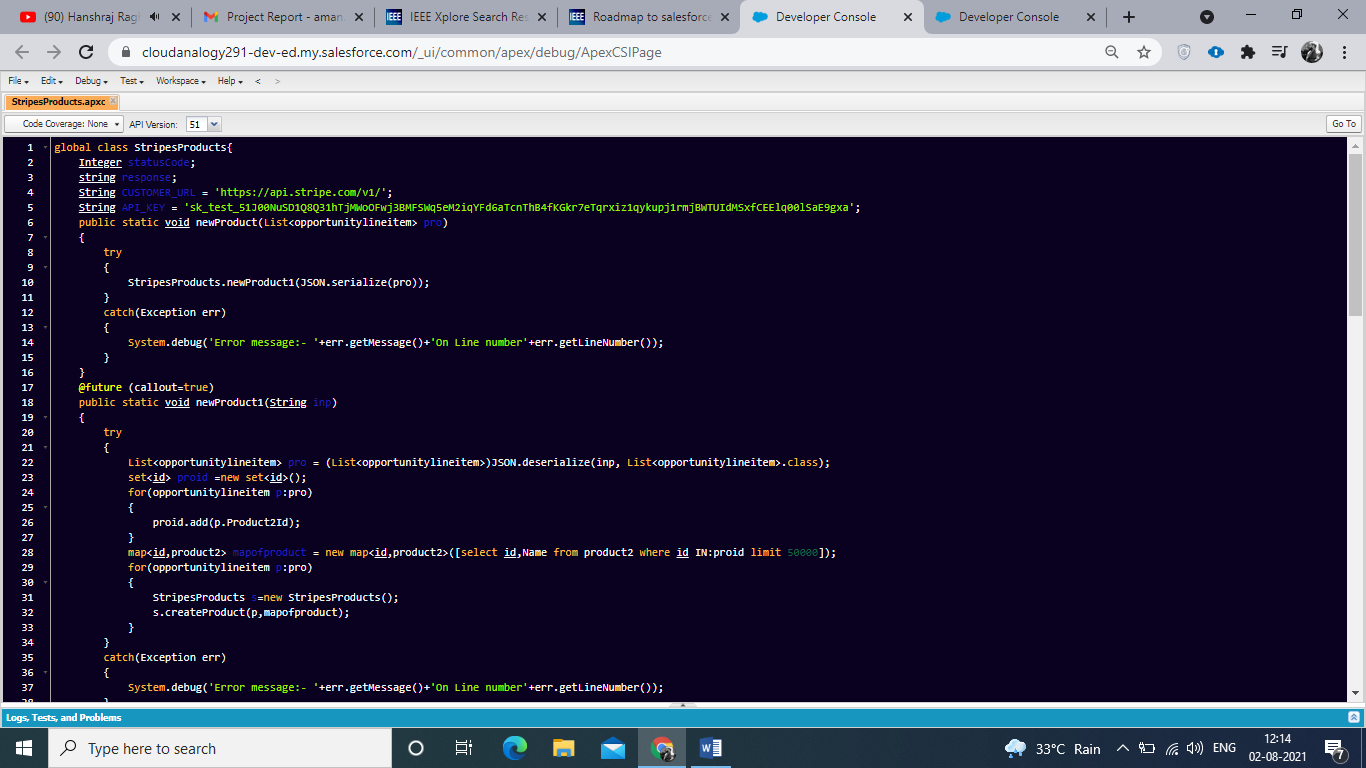


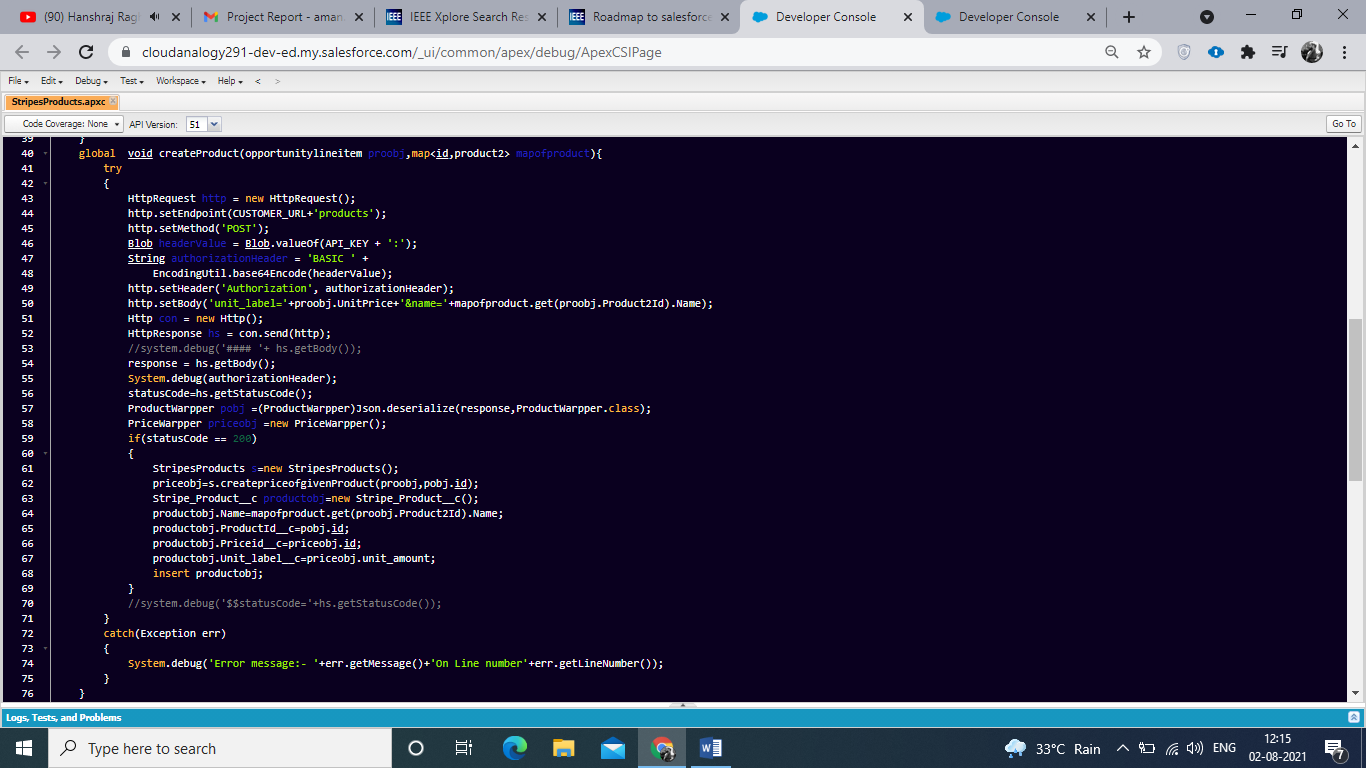


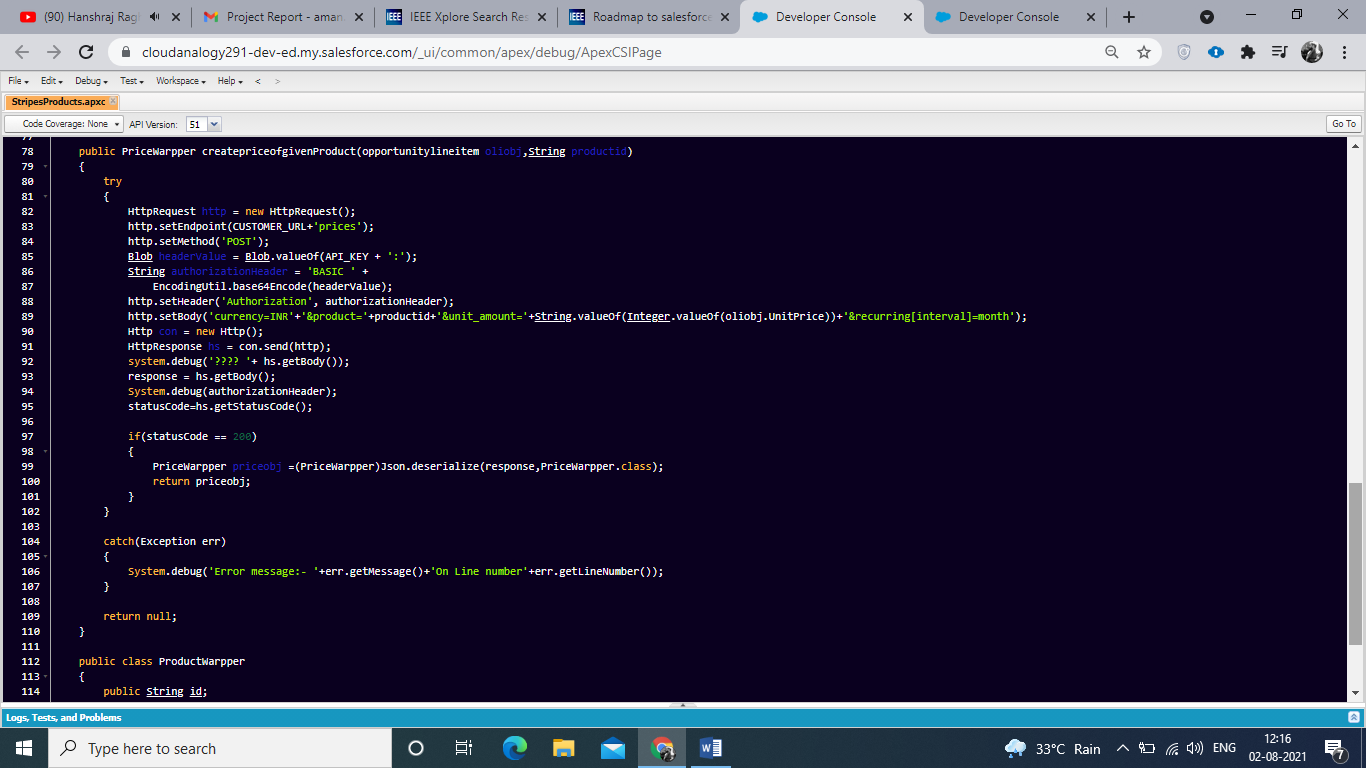
**4.1.5 Apex Trigger for create product.**



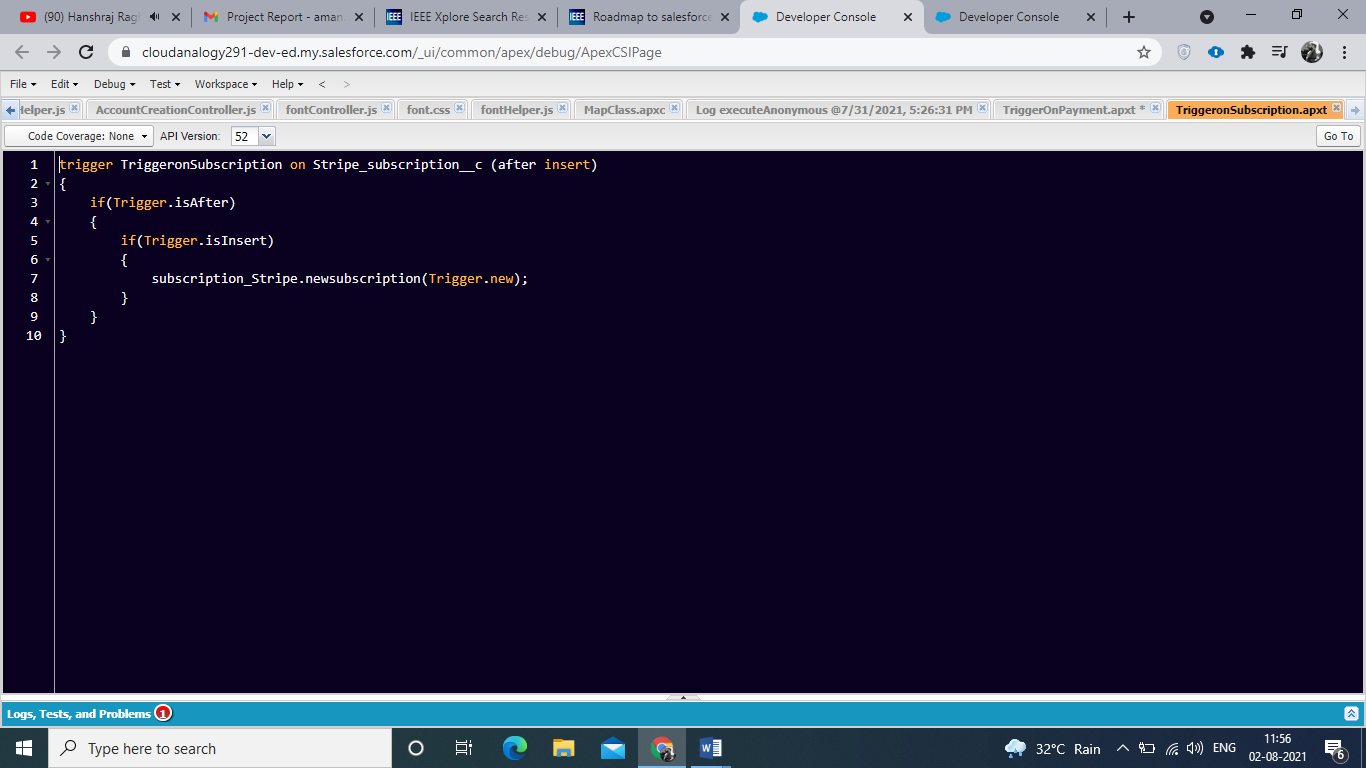
**4.1.6 Apex Class for create products.**



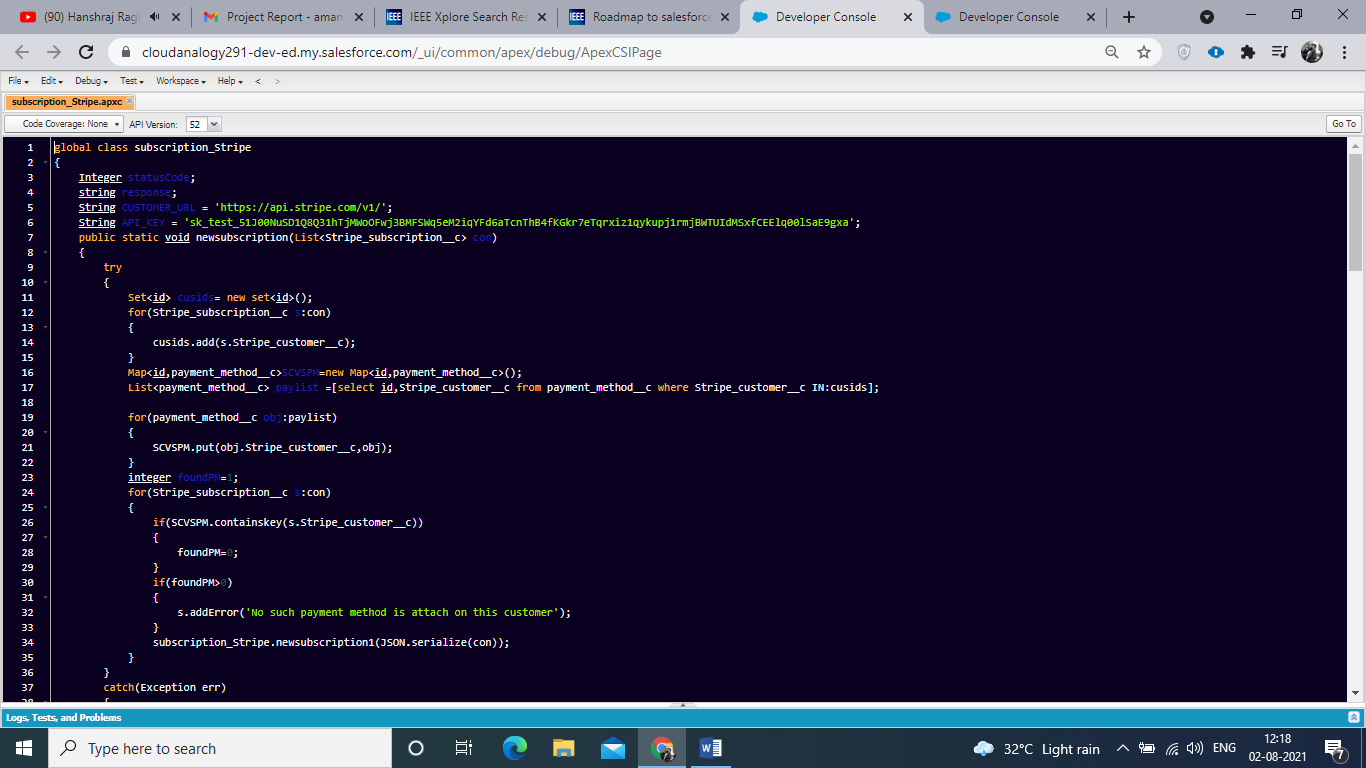


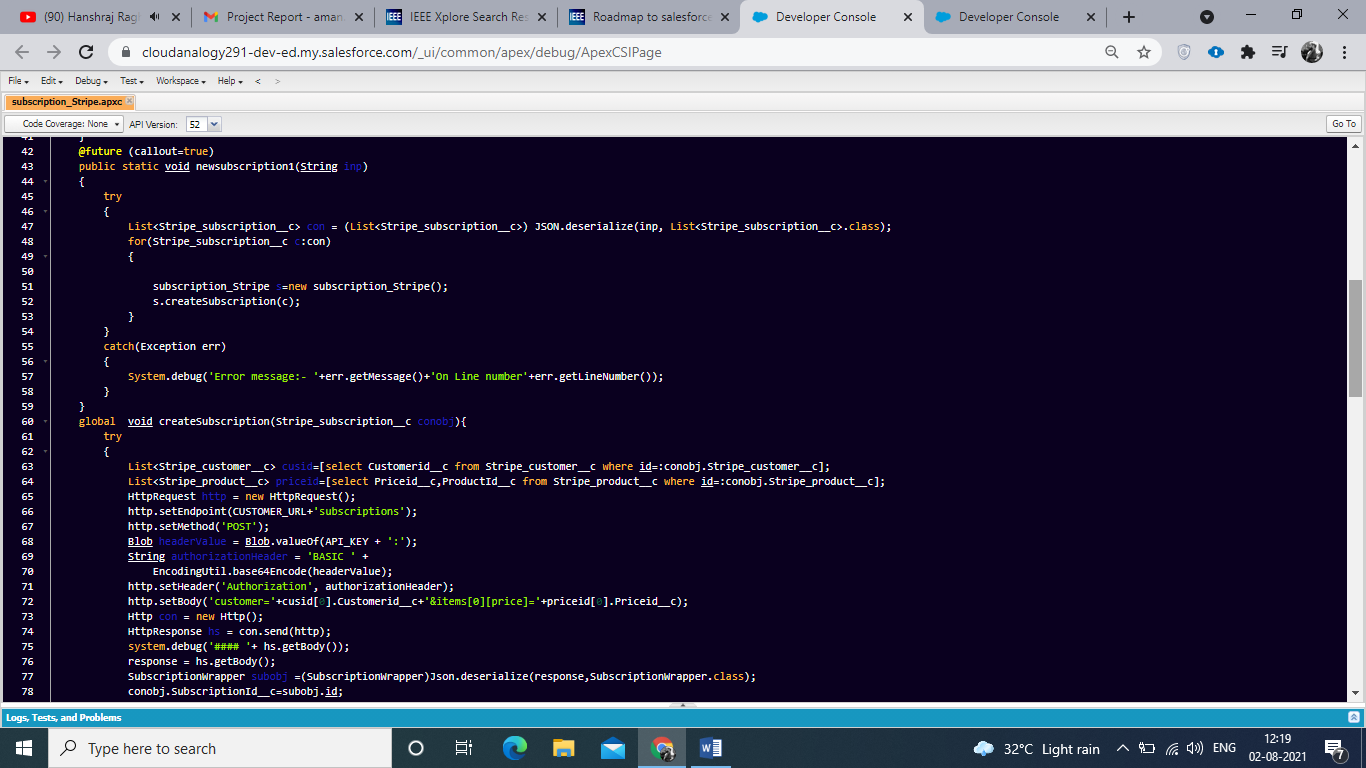


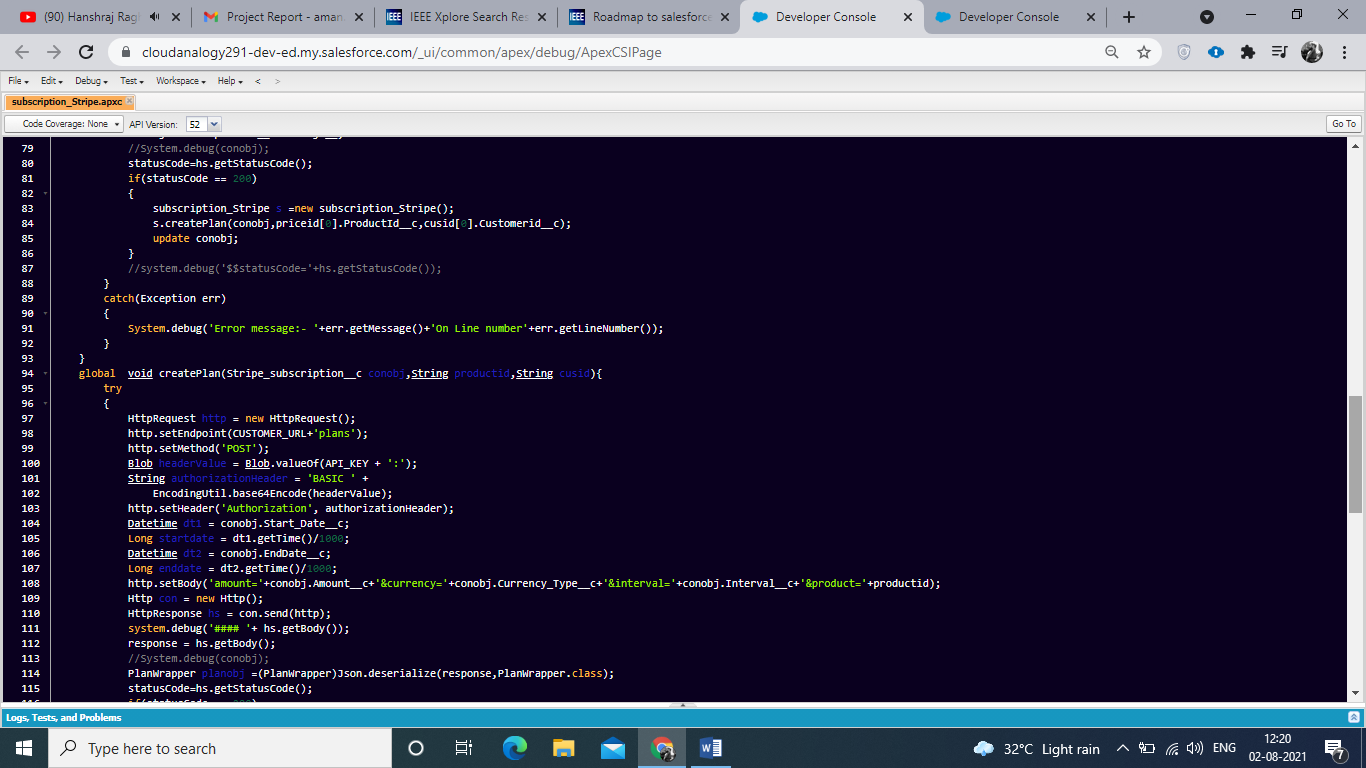
**4.1.7 Apex Trigger for create subscription on product.**

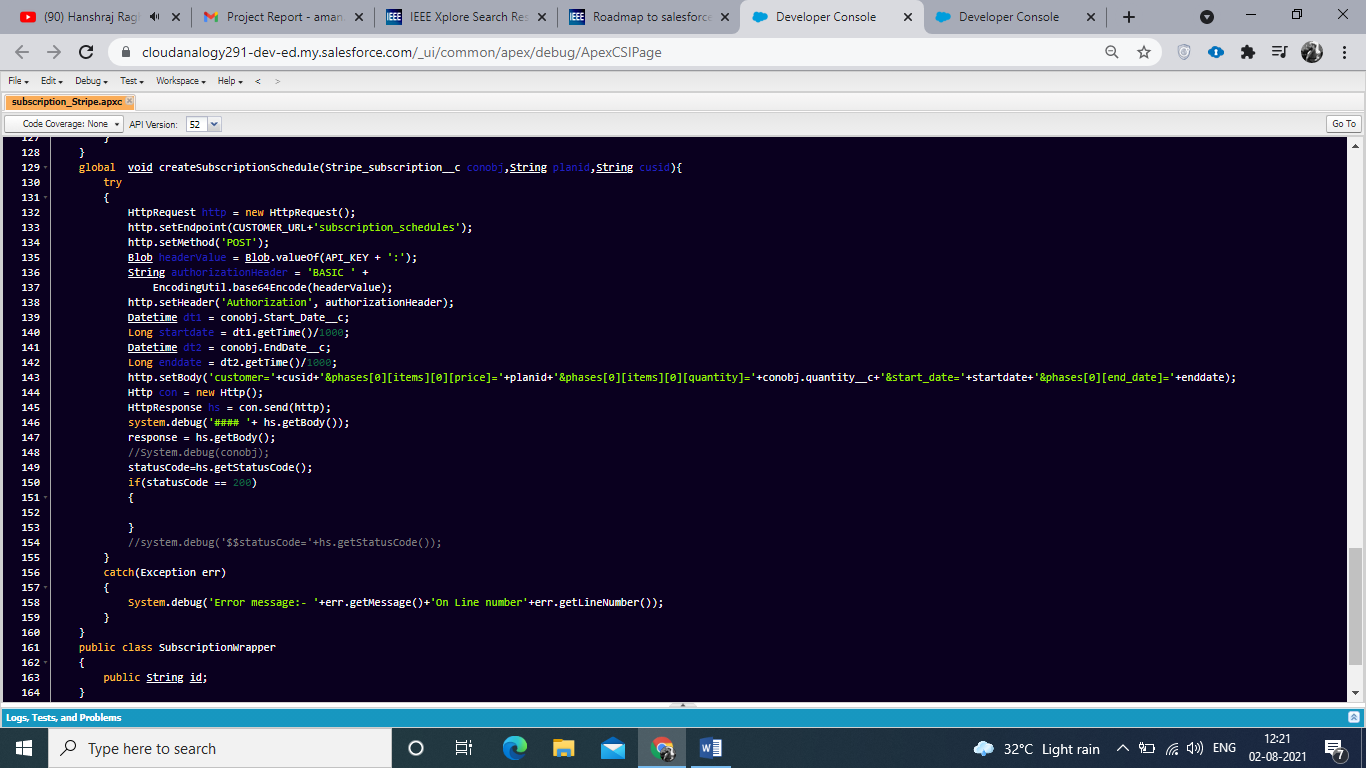


**4.1.8 Apex Class for create subscription on product.**

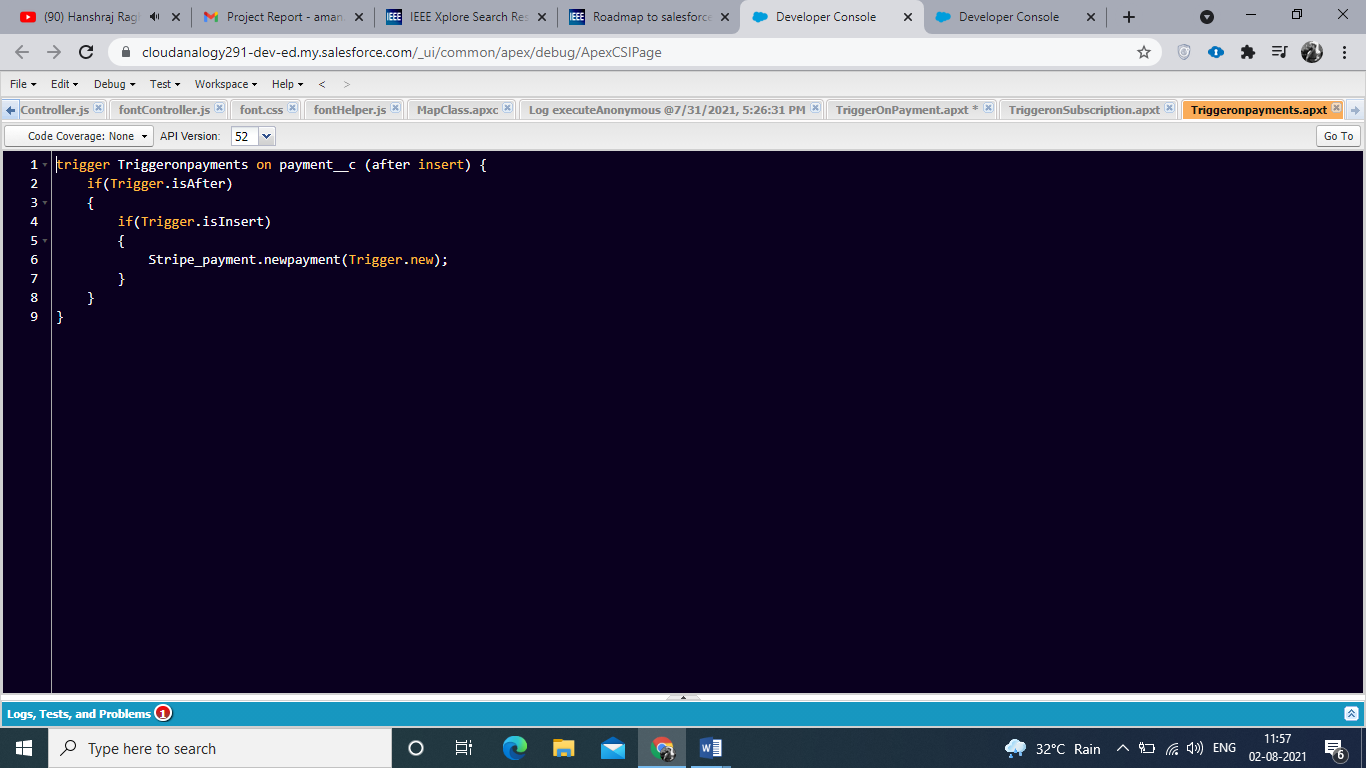




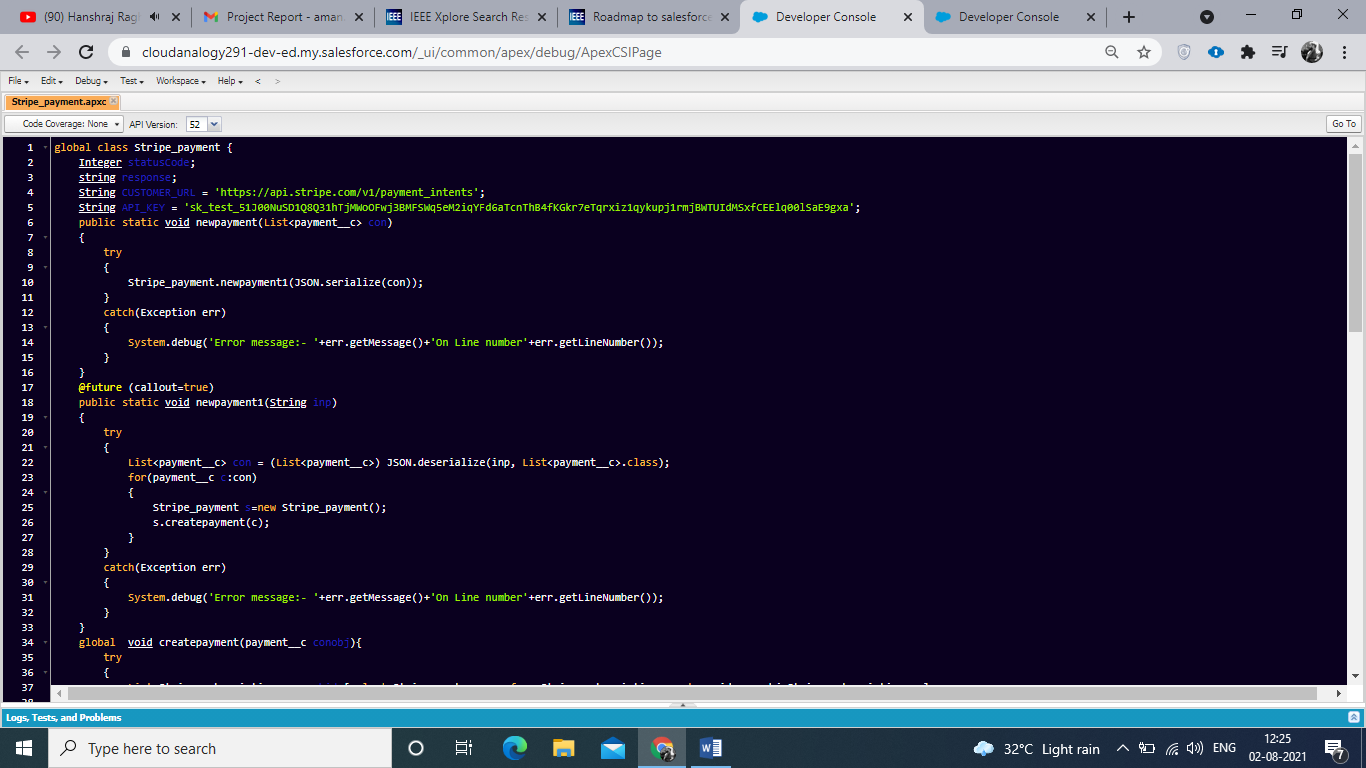




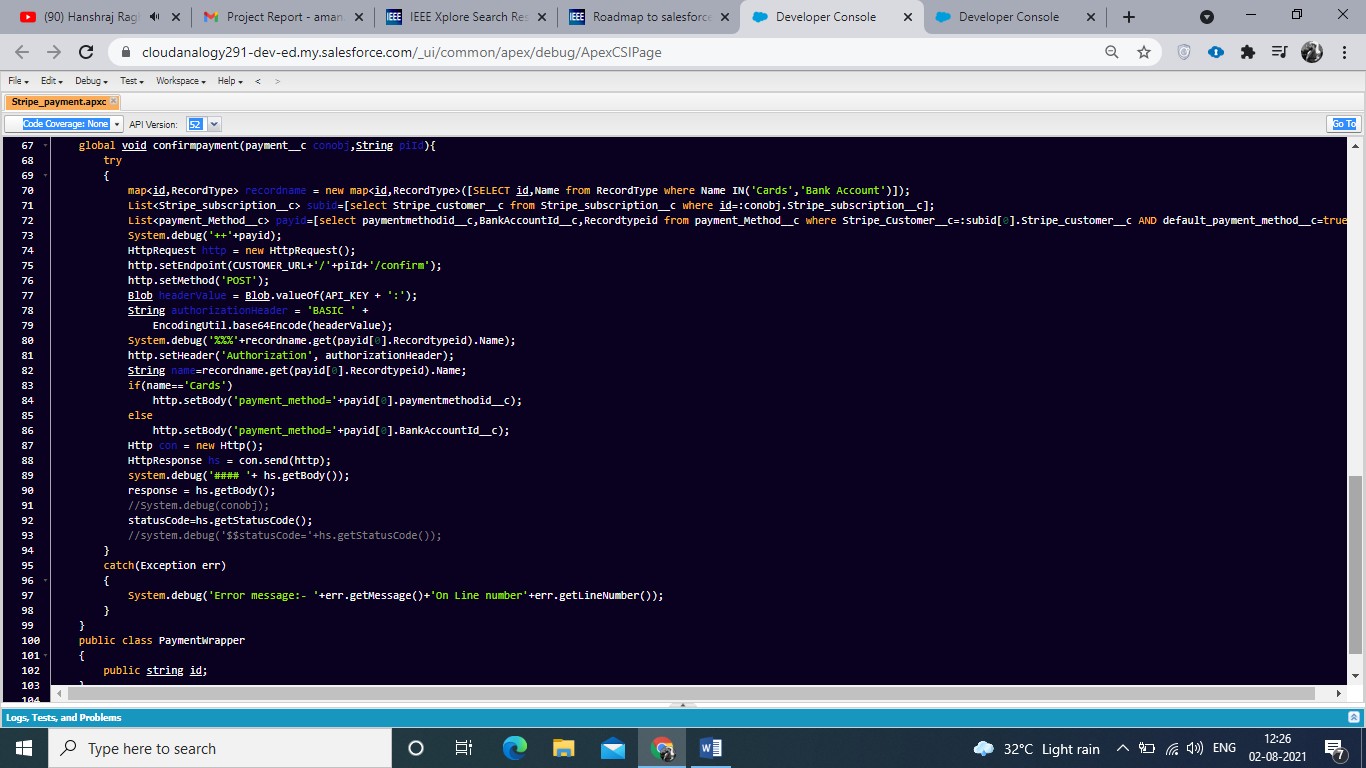
**4.1.9 Apex Trigger for create payment.**



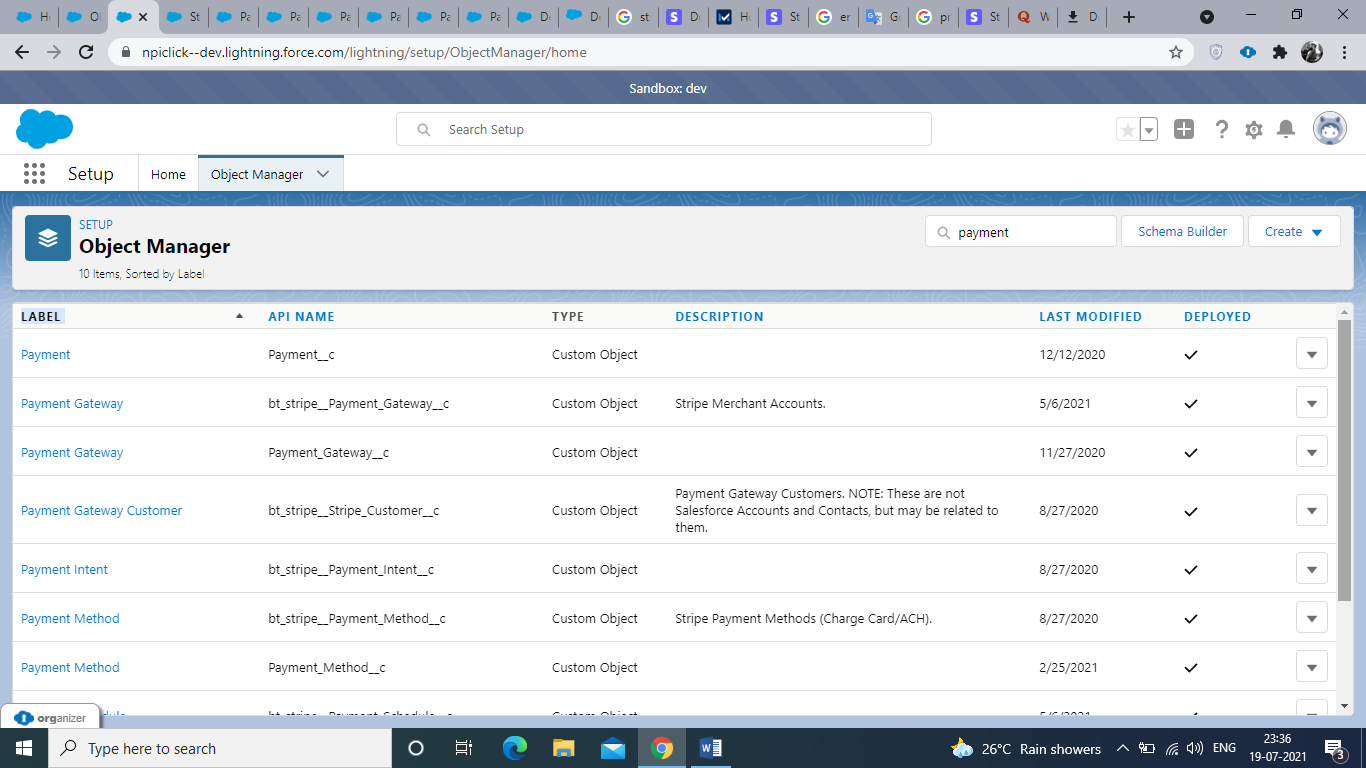
**4.1.10 Apex Class for create payment.**





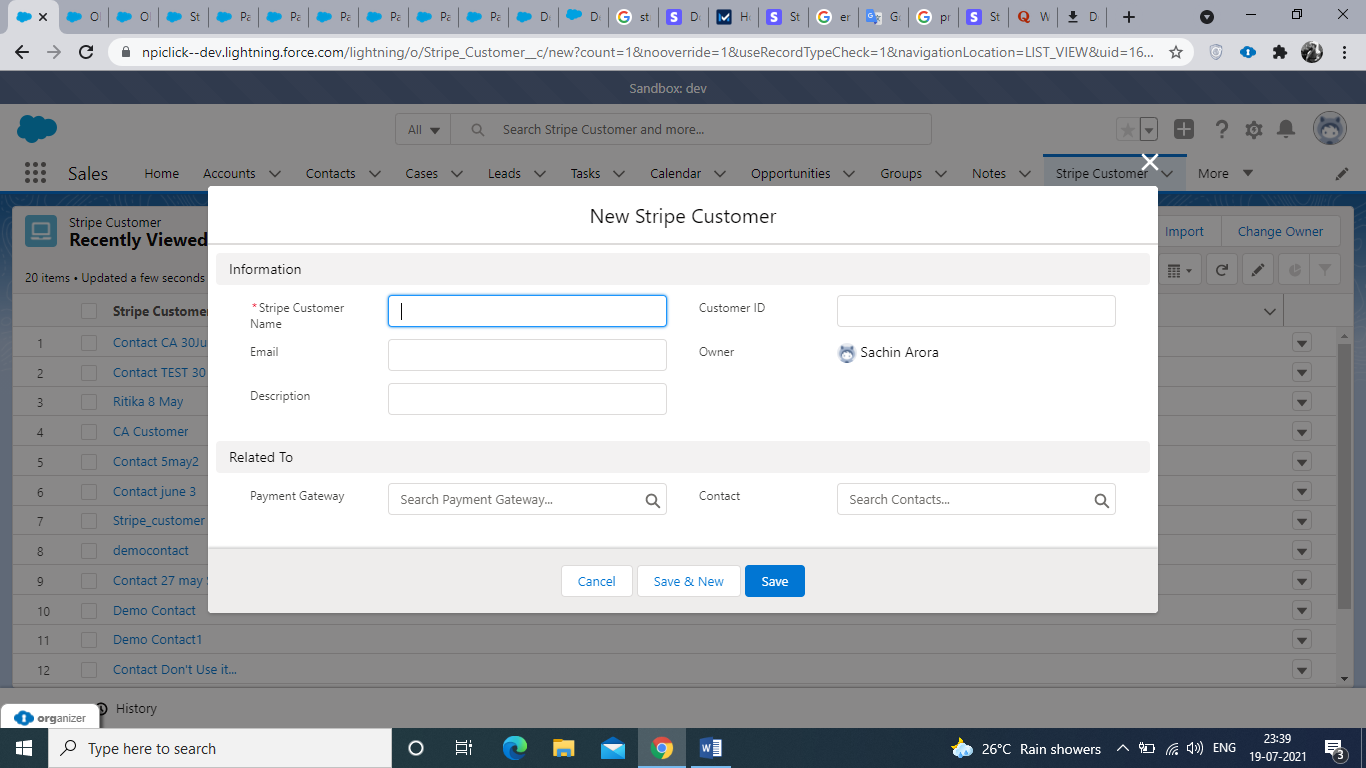


## 4.2 Screenshots of Modules: 4.2.1 Custom Objects Showing all custom objects.

****

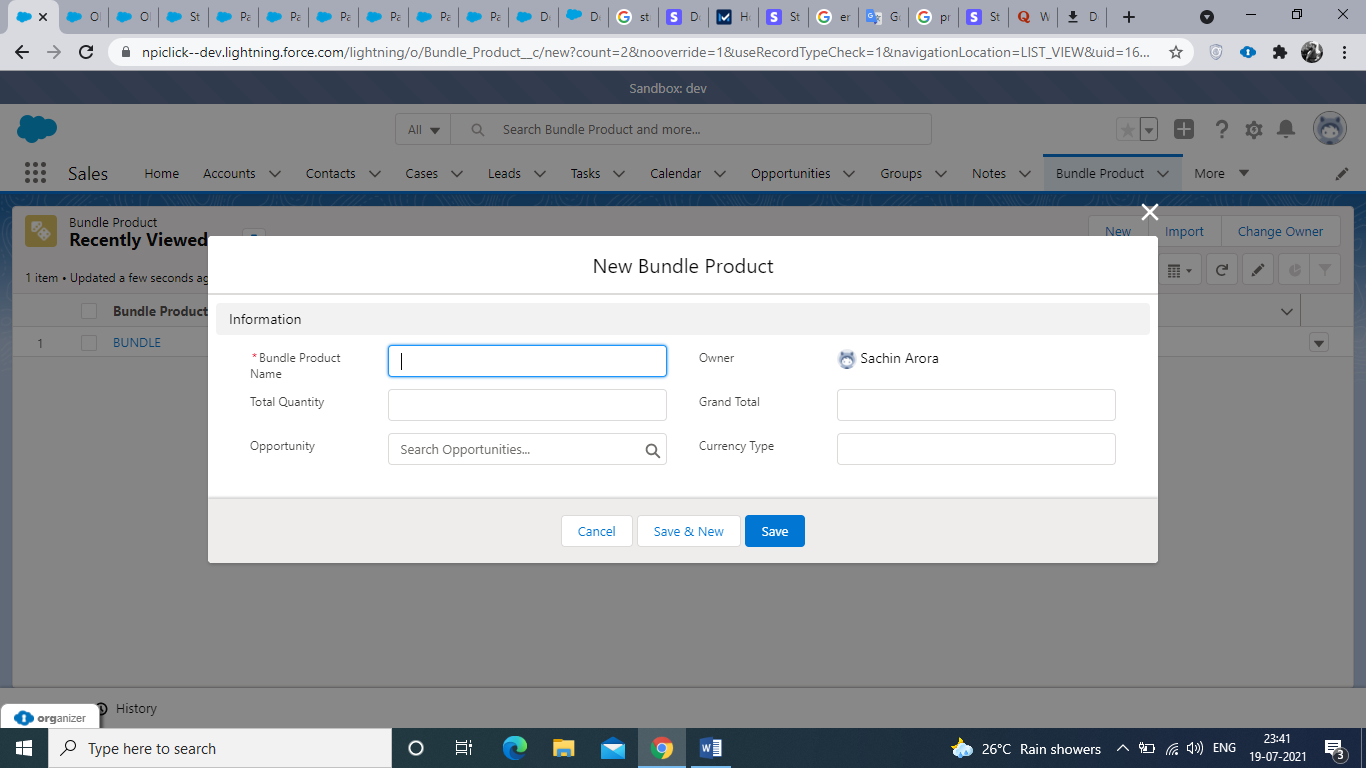
**Fig 4.1 Showing objects**

## 4.2.2 Customer Object: Customer objects allow you to perform recurring charges, and to track multiple charges, that are associated with the same customer. The API allows you to create, delete, and update your customers. You can retrieve individual customers as well as a list of all your customers.

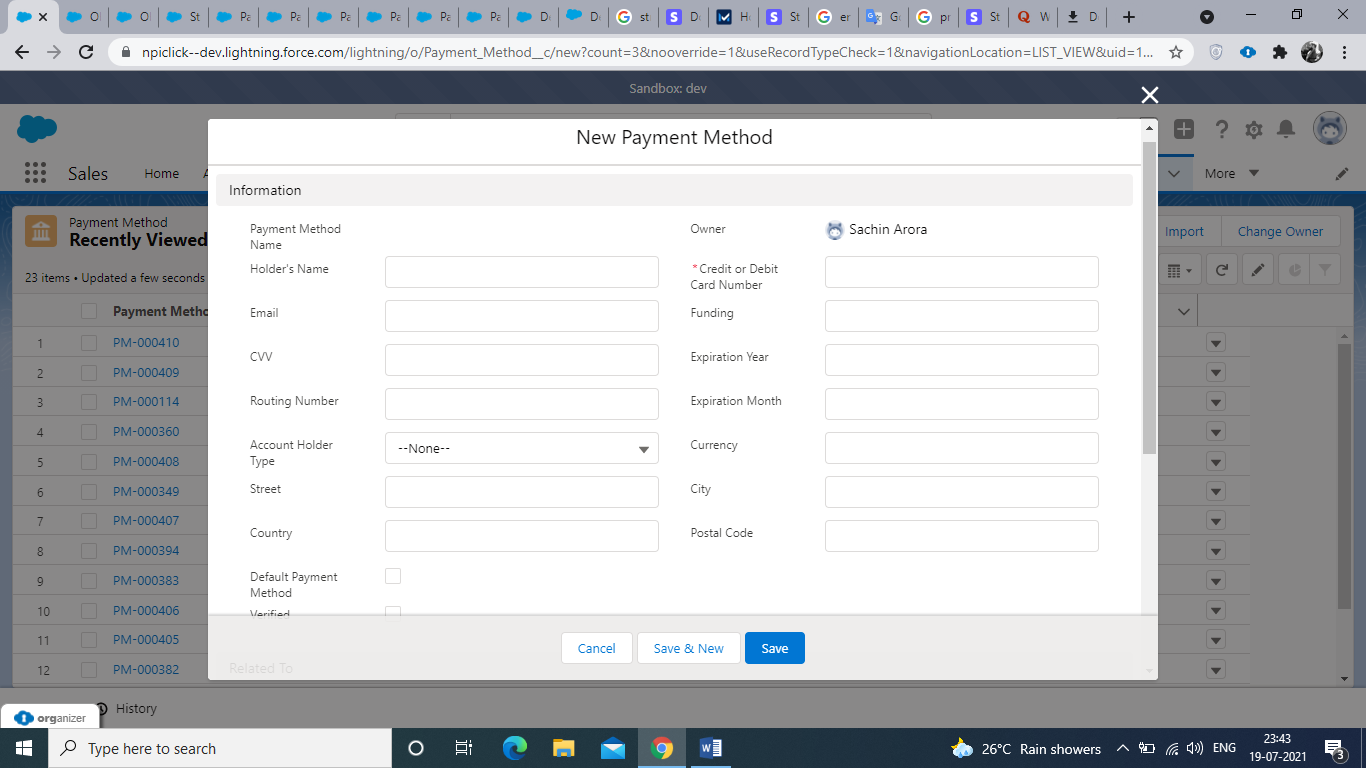


**Fig 4.2 Showing Customer** **Object**

**4.2.3 Product Object:**Products describe the specific goods or services you offer to your customers. For example, you might offer a Standard and Premium version of your goods or service; each version would be a separate Product. They can be used in conjunction with [Prices](https://stripe.com/docs/api/products#prices) to configure pricing in Checkout and Subscriptions and Prices define the unit cost, currency, and (optional) billing cycle for both recurring and one-time purchases of products. [Products](https://stripe.com/docs/api/prices#products) help you track inventory or provisioning, and prices help you track payment terms. Different physical goods or levels of service should be represented by products, and pricing options should be represented by prices. This approach lets you change prices without having to change your provisioning scheme.



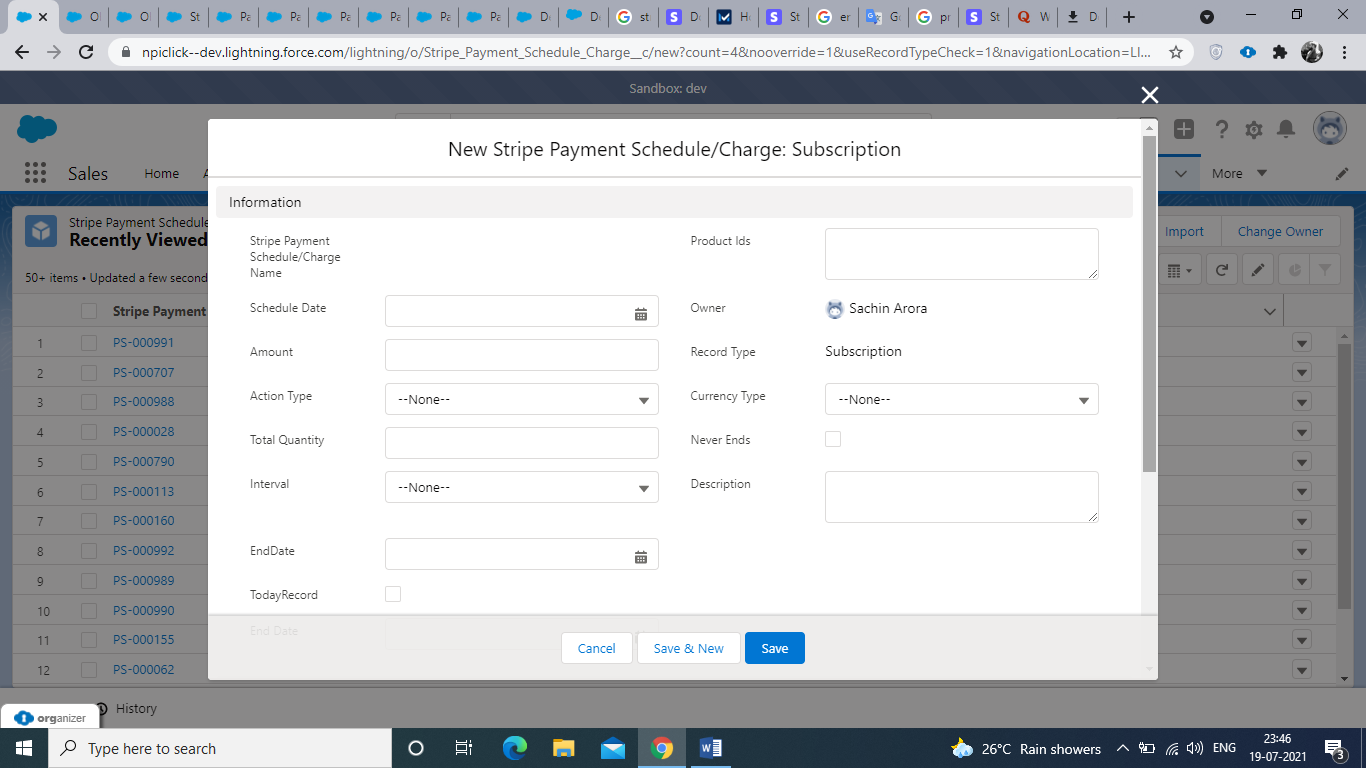
**Fig 4.3 Showing product object.**

**4.2.4 Payment Method object.**Payment Method objects represent your customer's payment instruments. They can be used with [Payment Intents](https://stripe.com/docs/payments/payment-intents) to collect payments or saved to Customer objects to store instrument details for future payments.

**Fig 4.4 Showing Payment method object.**

**4.2.5 Product Subscription Object:**

Subscriptions allow you to charge a customer on a recurring basis on product.

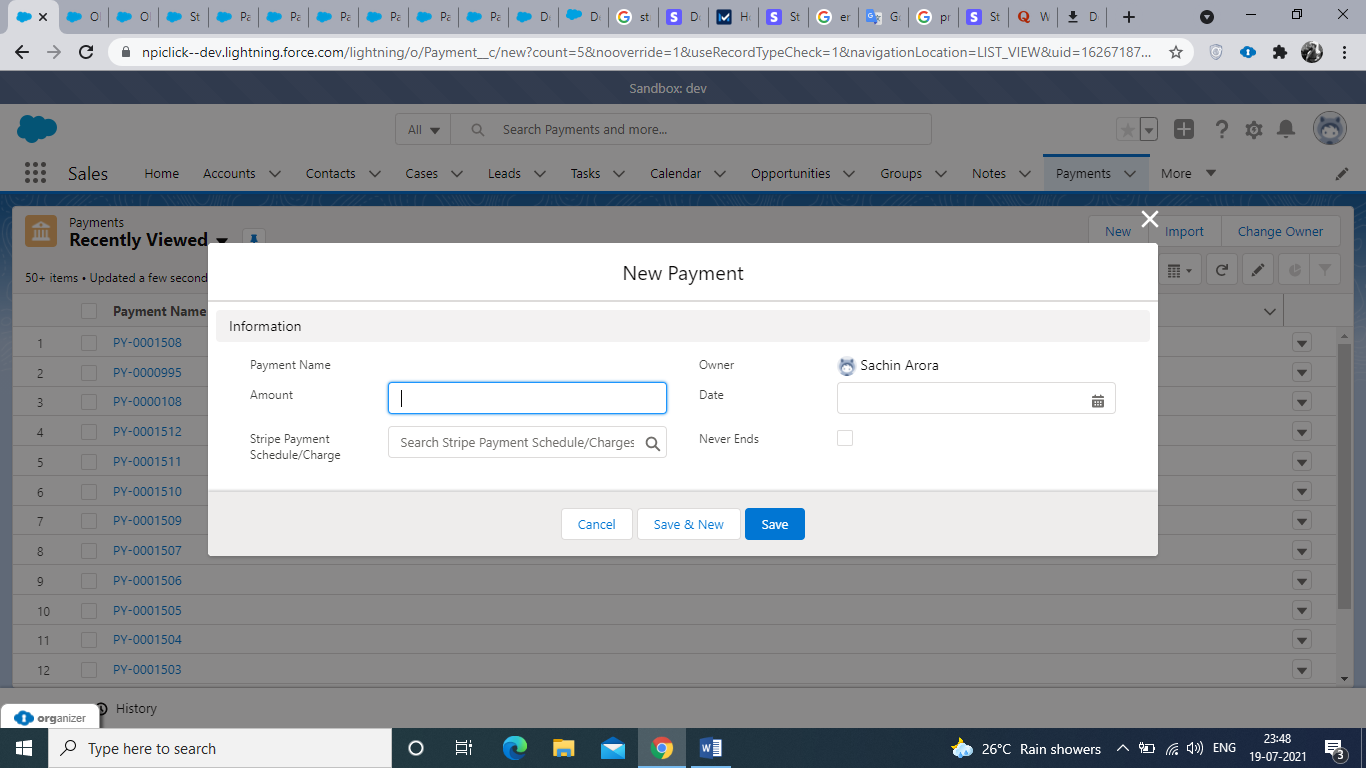


**Fig 4.5 Showing Product Subscription.**

**4.2.6 Payment Object:**

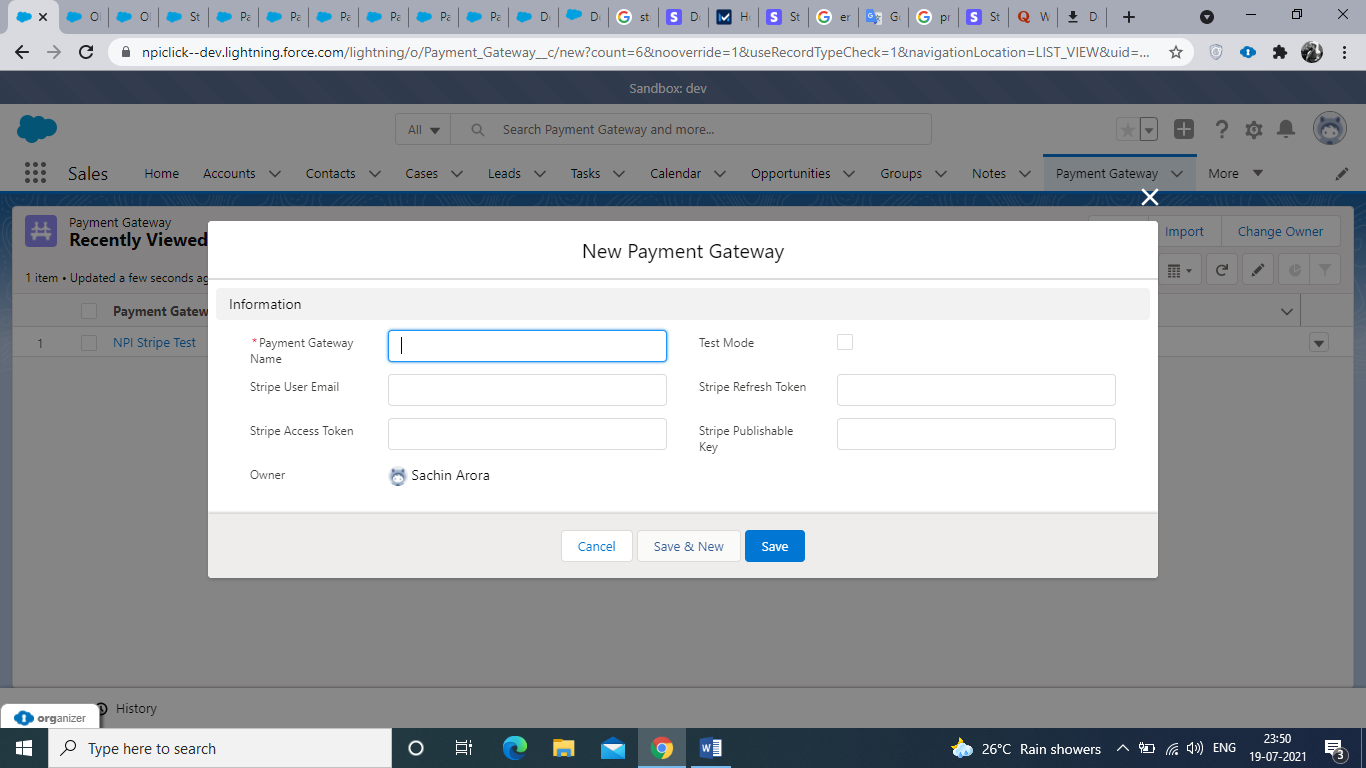
A Payment Intent guides you through the process of collecting a payment from your customer. We recommend that you create exactly one Payment Intent for each order or customer session in your system. You can reference the Payment Intent later to see the history of payment attempts for a particular session.

A Payment Intent transitions through [multiple statuses](https://stripe.com/docs/payments/intents#intent-statuses) throughout its lifetime as it interfaces with Stripe.js to perform authentication flows and ultimately creates at most one successful charge.



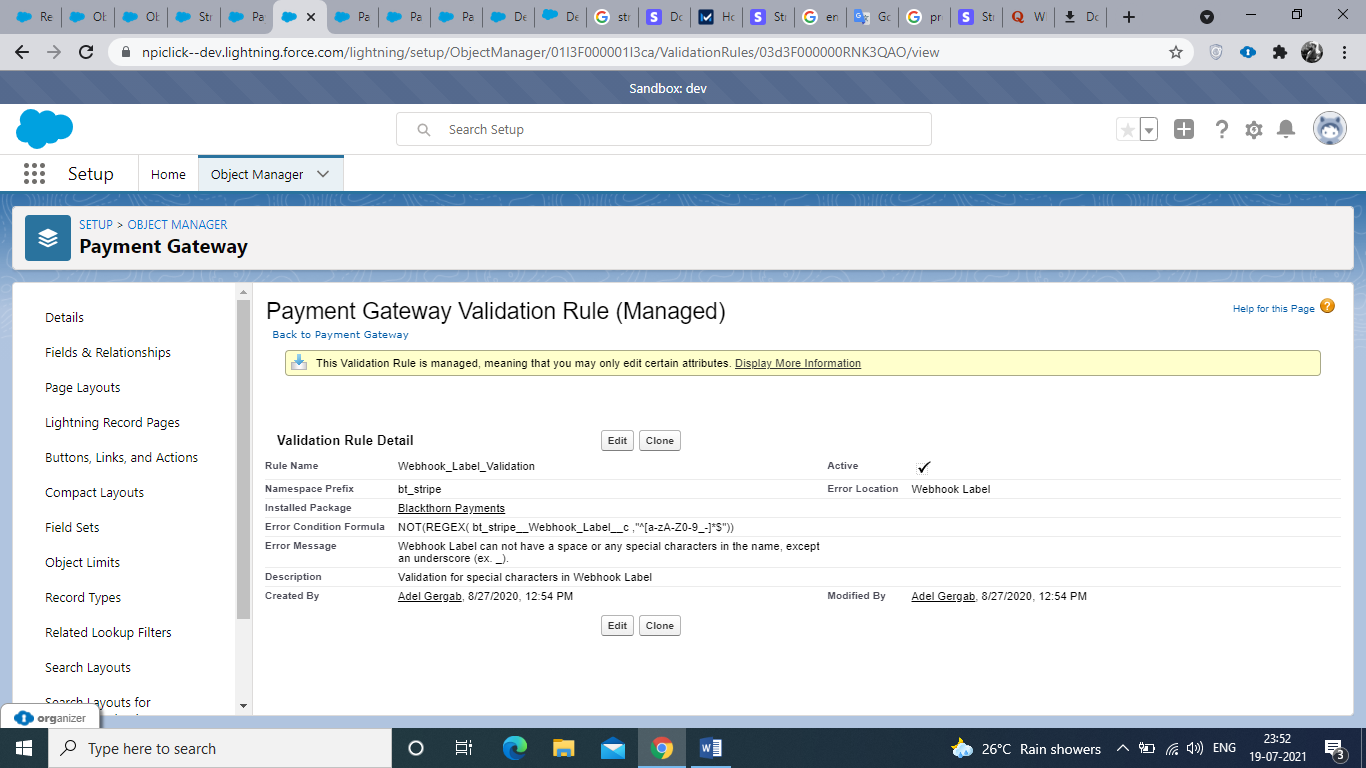
**Fig 4.6 Showing Payment Object.**

**4.2.7 Showing Payment Gateway Object.**You can add multiple payment gateway using this object and gateway are external service providers that process these electronic payments. Salesforce Billing uses out-of-the-box or API integrations to interface with a payment gateway.

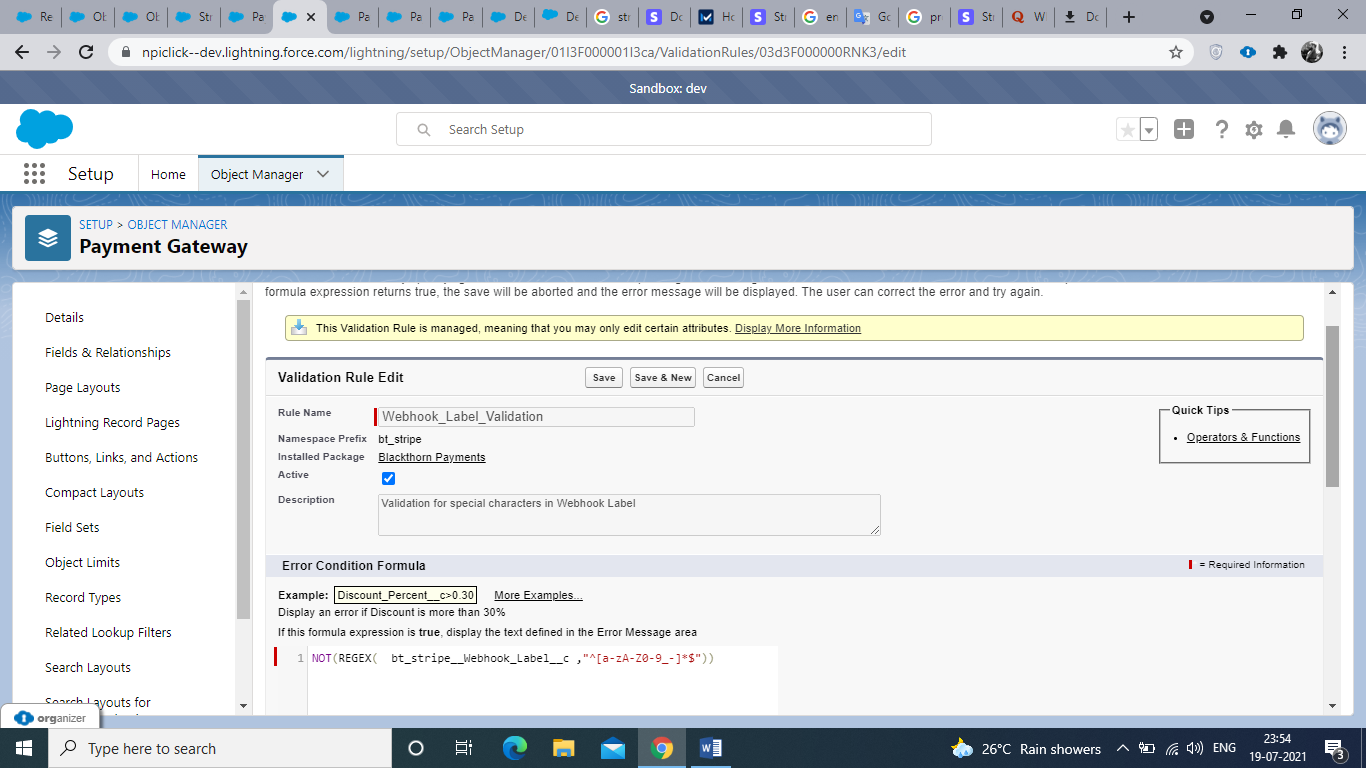


**Fig 4.7 Showing Payment Gateway Object.**

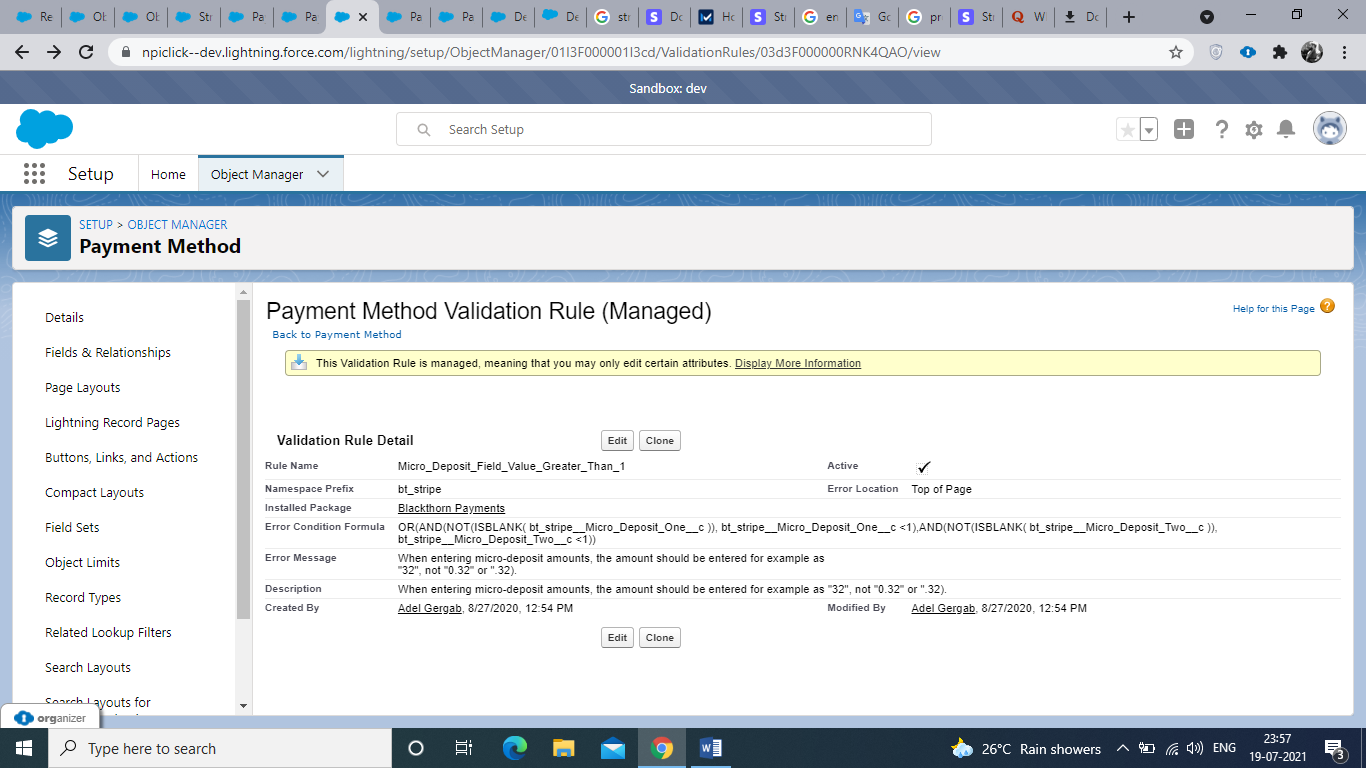
## 4.3 Validations On Object.



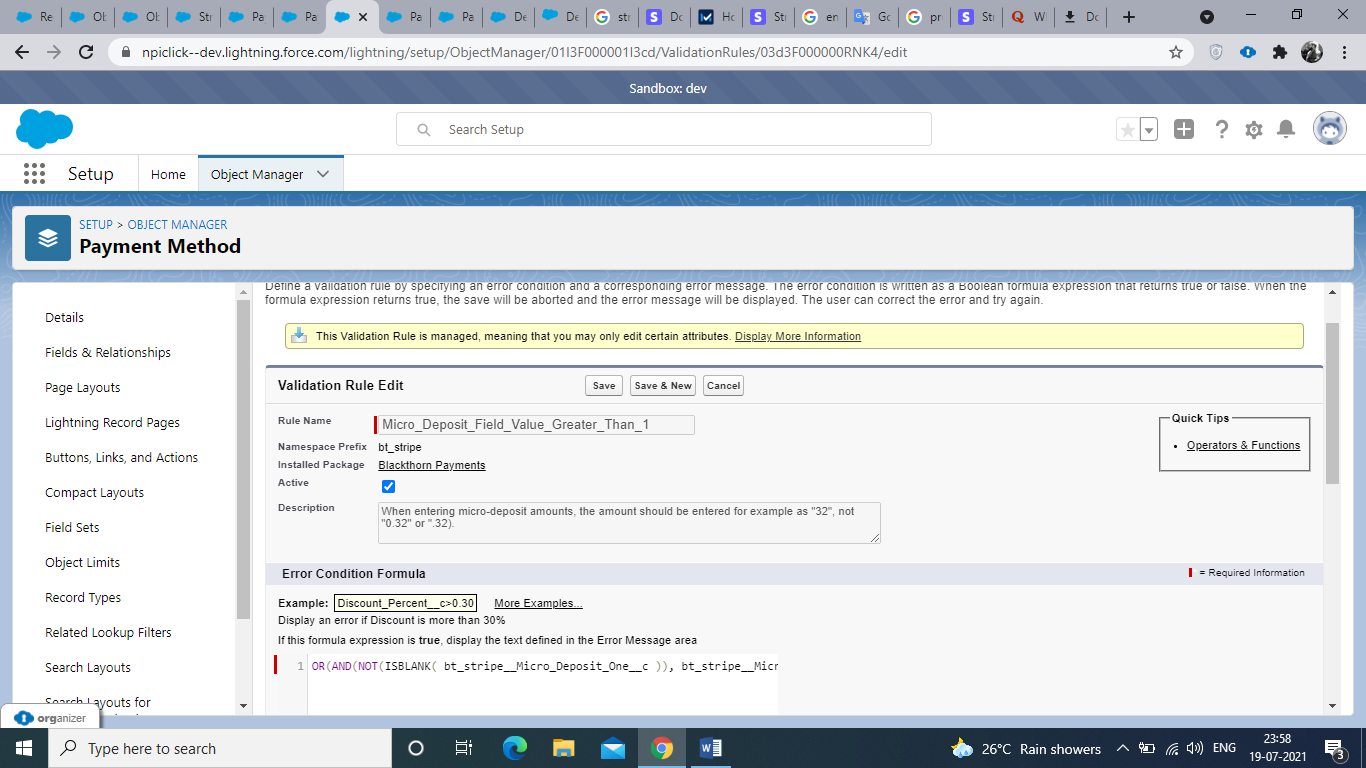
**Fig 4.8 Validations on Payment Gateway.**



**Fig 4.9 Validations on Payment Method.**



**Fig 4.10 Validations on Payment Object**



**Fig 4.11 Validations on Payment Object.**

**Chapter – 5**

**Testing**

The Force.com platform requires that at least 80% of the Apex Code in an org be executed via unit tests in order to deploy the code to production. You shouldn’t consider 80% code coverage to be an end ­goal though. Instead, you should strive to increase the state coverage of your unit tests. Code has many more possible states than it has lines of code. For example, the following method has 4,294,967,296 different states: System testing of software or hardware is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. System testing falls within the scope of black box testing, and as such, should require no knowledge of the inner design of the code or logic.

**5.1 TYPES OF TESTING**

**BLACK BOX TESTING:**

The technique of testing without having any knowledge of the interior workings of the application is called black ­box testing. The tester is oblivious to the system architecture and does not have access to the source code. Typically, while performing a black­ box test, a tester will interact with the system's user interface by providing inputs and examining outputs without knowing how and where the inputs are worked upon.

**WHITE BOX TESTING:**

White­box testing is the detailed investigation of internal logic and structure of the code. White­box testing is also called glass testing or open­ box testing. In order to perform white ­box testing on an application, a tester needs to know the internal workings of the code.

**GREY BOX TESTING**:

Grey ­box testing is a technique to test the application with having a limited knowledge of the internal workings of an application. In software testing, the phrase the more you know, the better carries a lot of weight while testing an application.

**UNIT TESTING:**

Unit Testing contains the testing of each unit of Recruitment Application. We have tested each interface by input values and check whether it is working properly working or not we also tested database connectivity. We have entered value in interface and check that the values are properly goes to corresponding tuples or not.

**INTEGRATION TESTING:**

Integration testing is defined as the testing of combined parts of an application to determine if they function correctly. Integration testing can be done in two ways: Bottom­ up integration testing and Top ­down integration testing.

**SYSTEM TESTING**:

System testing tests the system as a whole. Once all the components are integrated, the application as a whole is tested rigorously to see that it meets the specified Quality Standards. This type of testing is performed by a specialized testing team

**AUTOMATION:**

Automation Testing or Test Automation is a software testing technique that performs using special automated testing software tools to execute a test case suite. On the contrary, Manual Testing is performed by a human sitting in front of a computer carefully executing the test steps.

CHAPTER 6

CONCLUSION

**6.1 CONCLUSION**

The main agenda of this project was to meet client needs with in the given period of time. In this project, I have worked on different parts of Salesforce including Lightning Components, Lightning Components, Apex classes, triggers etc. Through all these tasks I have learned a lot of things about salesforce and its implementation and gained knowledge and skills. Integration with a third party app i.e. Stripe was the most interesting part for me as it was totally new thing for me and was working first time on integration and Webservices. In this whole process I have used

* Apex Language
* Java Script
* HTML
* Lightning Design System
* Cascading Style Sheets.
* Stripe (For integration)
* Triggers
* Webservices

**Chapter – 7**

**Bibliography**

* trailhead.salesforce.com
* developer.salesforce.com
* https://www.lightningdesignsystem.com/
* https://developer.salesforce.com/docs/component-library/overview/components
* Communityforce.com

## References

1. [Krutarth Soni](https://ieeexplore.ieee.org/author/37086259232);[Brijesh Vala](https://ieeexplore.ieee.org/author/37086252046)  
     
   First published:2017  
   ISBN: 978-1-5090-3239-6, 978-1-5090-3240-2
2. [Jigar Patel](https://ieeexplore.ieee.org/author/37086003992);[Ankit Chouhan](https://ieeexplore.ieee.org/author/37085997823)  
     
   First published:2017  
   ISBN: 978-1-5090-3239-6,978-1-5090-3240-2
3. Radhika Gupta;Sahil Verma;Kavita Janjua  
     
   First published:2018  
   ISBN: 978-1-5386-8025-4, 978-1-5386-8026-1
4. [Manvi Seth](https://ieeexplore.ieee.org/author/37086468122)  
     
   First published:2018  
   ISBN: 978-1-5386-5886-4, 978-1-5386-5887-1
5. Anuradha Manchar;[Ankit Chouhan](https://ieeexplore.ieee.org/author/37085997823)  
     
   First published:2017  
   ISBN: 978-1-5090-3239-6, 978-1-5090-3240-2
6. [Jing Hu](https://ieeexplore.ieee.org/author/37088507105);[Bo Zhang](https://ieeexplore.ieee.org/author/37088411502)  
     
   First published:2020  
   ISBN: 978-1-7281-6499-1, 978-1-7281-6500-4
7. [Aneta Poniszewska-Maranda](https://ieeexplore.ieee.org/author/38272891100);[Radosław Matusiak](https://ieeexplore.ieee.org/author/37086047150);[Natalia Kryvinska](https://ieeexplore.ieee.org/author/37297240400)  
     
   First published:2017  
   ISBN: 978-1-5386-2005-2, 978-1-5386-2004-5
8. [C. S. Lee](https://ieeexplore.ieee.org/author/37087414008);[A. Tiong](https://ieeexplore.ieee.org/author/37087411914);[W. L. Tang](https://ieeexplore.ieee.org/author/37087412823);[K. H. Yap](https://ieeexplore.ieee.org/author/37087411986)  
     
   First published:2019  
   ISBN: 978-1-7281-3804-6, 978-1-7281-3805-3
9. L.N. Goslin;C.L. Petersen  
     
   First published:1997  
   ISBN: 0-7803-3574-0
10. [Steve Fisher](https://ieeexplore.ieee.org/author/37690142000)  
      
    First published:2007  
    ISBN: 0-7695-2892-9
11. [Jigar Patel](https://ieeexplore.ieee.org/author/37086003992);[Ankit Chouhan](https://ieeexplore.ieee.org/author/37085997823)  
      
    First published:2016  
    ISBN: 978-1-5090-1066-0, 978-1-5090-1065-3, 978-1-5090-1067-7
12. [Amitabh Saxena](https://ieeexplore.ieee.org/author/37077744400);[Shubhashis Sengupta](https://ieeexplore.ieee.org/author/37553454200);[Pradeepkumar Duraisamy](https://ieeexplore.ieee.org/author/37077549100);[Vikrant Kaulgud](https://ieeexplore.ieee.org/author/38537630200);[Amit Chakraborty](https://ieeexplore.ieee.org/author/37077894000)  
      
    First published:2013  
    ISBN: 978-1-4673-6217-7, 978-1-4799-2432-5
13. [Sodam Baek](https://ieeexplore.ieee.org/author/37085377987);[Kibae Kim](https://ieeexplore.ieee.org/author/37406159900);[Jörn Altmann](https://ieeexplore.ieee.org/author/37270584900)  
      
    First published:2014  
    ISBN: 978-1-4799-5779-8
14. [Andrea Leszek](https://ieeexplore.ieee.org/author/38094545700);[Catherine Courage](https://ieeexplore.ieee.org/author/38094545200)  
      
    First published:2008  
    ISBN: 978-1-4799-5779-8
15. [Louise Ann Lyon](https://ieeexplore.ieee.org/author/37085658259);[Kieren Jameson](https://ieeexplore.ieee.org/author/37085673610)  
      
    First published:2015  
    ISBN: 978-1-4673-7457-6