

SNJB's Late Sau. K. B. Jain College of Engineering, Chandwad

Department of Computer Engineering

Course Name:Laboratory Practice II(310258):Cloud Computing

Class:Third Year (TE) Div A/ Div B

Batch:T1/T2/T3/T4

Name:

Roll No:

Assignment No: 7

Answers (A) – 5M	Coding Efficiency (C) – 5M	Viva (V) – 5M	Timely Completion (T) – 5M	Total(20M)	Sign

Date of Performance:..... **Date of Completion:**.....

1. Title of Assignment:

Case study on Microsoft azure to learn about Microsoft Azure is a cloud computing platform and infrastructure, created by Microsoft, for building, deploying and managing applications and services through a global network of Microsoft-managed data centers.

2. Objective:

1. Study of Microsoft Azure Cloud Platform
2. Create a static HTML web app in Azure

3. Outcome: Use tools and techniques in the area of Cloud Computing

4. Software and Hardware Requirement:

Software Requirement: Latest Browser, www.azure.com

Hardware Requirement: Internet Connection, PC with Min. 2GB RAM, Core i5 Processor

5.Relevant Theory :

What is Azure?

- Azure is a complete cloud platform that can host your existing applications and streamline new application development.

- Azure can even enhance on-premises applications.
- Azure integrates the cloud services that you need to develop, test, deploy, and manage your applications, all while taking advantage of the efficiencies of cloud computing.
- By hosting your applications in Azure, you can start small and easily scale your application as your customer demand grows.
- Azure also offers the reliability that's needed for high-availability applications, even including failover between different regions.
- The [Azure portal](#) lets you easily manage all your Azure services.
- You can also manage your services programmatically by using service-specific APIs and templates.
- This guide is an introduction to the Azure platform for application developers.
- It provides guidance and direction that you need to start building new applications in Azure or migrating existing applications to Azure.

Where do I start?

- With all the services that Azure offers, it can be an intimidating task to figure out which services you need to support your solution architecture.
- This section highlights the Azure services that developers commonly use.
- For a list of all Azure services, see the [Azure documentation](#).
- First, you must decide on how to host your application in Azure.
- Do you need to manage your entire infrastructure as a virtual machine (VM)? Can you use the platform management facilities that Azure provides? Maybe you need a serverless framework to host code execution only?
- Your application needs cloud storage, which Azure provides several options for. You can take advantage of Azure's enterprise authentication.
- There are also tools for cloud-based development and monitoring, and most hosting services offer DevOps integration.

Now, let's look at some of the specific services that we recommend investigating for your applications.

Application hosting

- Azure provides several cloud-based compute offerings to run your application so that you don't have to worry about the infrastructure details.
- You can easily scale up or scale out your resources as your application usage grows.
- Azure offers services that support your application development and hosting needs.

- Azure provides Infrastructure as a Service (IaaS) to give you full control over your application hosting.
- Azure's Platform as a Service (PaaS) offerings provide the fully managed services needed to power your apps.
- There's even true serverless hosting in Azure where all you need to do is write your code.

Azure App Service

- When you want the quickest path to publish your web-based projects, consider Azure App Service.
- App Service makes it easy to extend your web apps to support your mobile clients and publish easily consumed REST APIs.
- This platform provides authentication by using social providers, traffic-based auto scaling, testing in production, and continuous and container-based deployments.

You can create web apps, mobile app back ends, and API apps.

- Because all three app types share the App Service runtime, you can host a website, support mobile clients, and expose your APIs in Azure, all from the same project or solution.
- To learn more about App Service, see

What is Azure Web Apps.

- App Service has been designed with DevOps in mind.
- It supports various tools for publishing and continuous integration deployments.
- These tools include GitHub webhooks, Jenkins, Azure DevOps, TeamCity, and others.
- You can migrate your existing applications to App Service by using the [online migration tool](#).
- **When to use:** Use App Service when you're migrating existing web applications to Azure, and when you need a fully-managed hosting platform for your web apps.
- You can also use App Service when you need to support mobile clients or expose REST APIs with your app.

Get started: App Service makes it easy to create and deploy your first [web app](#), [mobile app](#), or [API app](#).

Try it now: App Service lets you provision a short-lived app to try the platform without having to sign up for an Azure account. Try the platform and [create your Azure App Service app](#).

Azure Virtual Machines

- As an Infrastructure as a Service (IaaS) provider, Azure lets you deploy to or migrate your application to either Windows or Linux VMs.
- Together with Azure Virtual Network, Azure Virtual Machines supports the deployment of Windows or Linux VMs to Azure.
- With VMs, you have total control over the configuration of the machine.
- When using VMs, you're responsible for all server software installation, configuration, maintenance, and operating system patches.
- Because of the level of control that you have with VMs, you can run a wide range of server workloads on Azure that don't fit into a PaaS model.
- These workloads include database servers, Windows Server Active Directory, and Microsoft SharePoint.

When to use: Use Virtual Machines when you want full control over your application infrastructure or to migrate on-premises application workloads to Azure without having to make changes.

- Create a [Linux VM](#) or [Windows VM](#) from the Azure portal.

Azure Functions (serverless)

- Rather than worrying about building out and managing a whole application or the infrastructure to run your code, what if you could just write your code and have it run in response to events or on a schedule? [Azure Functions](#) is a "serverless"-style offering that lets you write just the code you need.
- With Functions, you can trigger code execution with HTTP requests, webhooks, cloud service events, or on a schedule.
- You can code in your development language of choice, such as C#, F#, Node.js, Python, or PHP. With consumption-based billing, you pay only for the time that your code executes, and Azure scales as needed.

When to use: Use Azure Functions when you have code that is triggered by other Azure services, by web-based events, or on a schedule.

You can also use Functions when you don't need the overhead of a complete hosted project or when you only want to pay for the time that your code runs.

To learn more, see [Azure Functions Overview](#).

- Follow the Functions quickstart tutorial to [create your first function](#) from the portal.
- Azure Functions lets you run your code without having to sign up for an Azure account. Try it now and [create your first Azure Function](#).

Azure Service Fabric

- Azure Service Fabric is a distributed systems platform.
- This platform makes it easy to build, package, deploy, and manage scalable and reliable microservices.
- It also provides comprehensive application management capabilities such as:
- Provisioning
- Deploying
- Monitoring
- Upgrading/Patching
- Deleting
- Apps, which run on a shared pool of machines, can start small and scale to hundreds or thousands of machines as needed.
- Service Fabric supports WebAPI with Open Web Interface for .NET (OWIN) and ASP.NET Core.
- It provides SDKs for building services on Linux in both .NET Core and Java.
- To learn more about Service Fabric, see the [Service Fabric documentation](#).

When to use: Service Fabric is a good choice when you're creating an application or rewriting an existing application to use a microservice architecture. Use Service Fabric when you need more control over, or direct access to, the underlying infrastructure.

Get started: [Create your first Azure Service Fabric application](#).

Azure Spring Cloud

- Azure Spring Cloud is a serverless microservices platform that enables you to build, deploy, scale and monitor your applications in the cloud.
- Use Spring Cloud to bring modern microservice patterns to Spring Boot apps, eliminating boilerplate code to quickly build robust Java apps.
 - Leverage managed versions of Spring Cloud Service Discovery and Config Server, while we ensure those critical components are running in optimum conditions.

- Focus on building your business logic and we will take care of your service runtime with security patches, compliance standards and high availability.
- Manage application lifecycle (for example, deploy, start, stop, scale) on top of Azure Kubernetes Service.
- Easily bind connections between your apps and Azure services such as Azure Database for MySQL and Azure Cache for Redis.
- Monitor and troubleshoot microservices and applications using enterprise-grade unified monitoring tools that offer deep insights on application dependencies and operational telemetry.

When to use: As a fully managed service Azure Spring Cloud is a good choice when you're minimizing operational cost running Spring Boot/Spring Cloud based microservices on Azure.

Get started: [Deploy your first Spring Boot app in Azure Spring Cloud.](#)

Enhance your applications with Azure services

Along with application hosting, Azure provides service offerings that can enhance the functionality. Azure can also improve the development and maintenance of your applications, both in the cloud and on-premises.

Hosted storage and data access

Most applications must store data, so however you decide to host your application in Azure, consider one or more of the following storage and data services.

- **Azure Cosmos DB:** A globally distributed, multi-model database service. This database enables you to elastically scale throughput and storage across any number of geographical regions with a comprehensive SLA.

When to use: When your application needs document, table, or graph databases, including MongoDB databases, with multiple well-defined consistency models.

- **Azure Storage:** Offers durable, highly available storage for blobs, queues, files, and other kinds of nonrelational data. Storage provides the storage foundation for VMs.

When to use: When your app stores nonrelational data, such as key-value pairs (tables), blobs, files shares, or messages (queues).

- **Azure SQL Database:** An Azure-based version of the Microsoft SQL Server engine for storing relational tabular data in the cloud. SQL Database provides predictable performance, scalability with no downtime, business continuity, and data protection.

When to use: When your application requires data storage with referential integrity, transactional support, and support for TSQL queries.

- You can use [Azure Data Factory](#) to move existing on-premises data to Azure.
- If you aren't ready to move data to the cloud, [Hybrid Connections](#) in Azure App Service lets you connect your App Service hosted app to on-premises resources.
- You can also connect to Azure data and storage services from your on-premises applications.

Docker support

- Docker containers, a form of OS virtualization, let you deploy applications in a more efficient and predictable way.
- A containerized application works in production the same way as on your development and test systems.
- You can manage containers by using standard Docker tools.
- You can use your existing skills and popular open-source tools to deploy and manage container-based applications on Azure.

Azure provides several ways to use containers in your applications.

- **Azure Kubernetes Service:** Lets you create, configure, and manage a cluster of virtual machines that are preconfigured to run containerized applications. To learn more about Azure Kubernetes Service, see [Azure Kubernetes Service introduction](#).

When to use: When you need to build production-ready, scalable environments that provide additional scheduling and management tools, or when you're deploying a Docker Swarm cluster.

Get started: [Deploy a Kubernetes Service cluster](#).

- **Docker Machine:** Lets you install and manage a Docker Engine on virtual hosts by using docker-machine commands.

When to use: When you need to quickly prototype an app by creating a single Docker host.

- **Custom Docker image for App Service:** Lets you use Docker containers from a container registry or a customer container when you deploy a web app on Linux.

When to use: When deploying a web app on Linux to a Docker image.

Get started: [Use a custom Docker image for App Service on Linux](#).

Authentication

- It's crucial to not only know who is using your applications, but also to prevent unauthorized access to your resources. Azure provides several ways to authenticate your app clients.
 - **Azure Active Directory (Azure AD):** The Microsoft multitenant, cloud-based identity and access management service. You can add single-sign on (SSO) to your applications by integrating with Azure AD. You can access directory properties by using the Microsoft Graph API. You can integrate with Azure AD support for the OAuth2.0 authorization framework and Open ID Connect by using native HTTP/REST endpoints and the multiplatform Azure AD authentication libraries.

When to use: When you want to provide an SSO experience, work with Graph-based data, or authenticate domain-based users.

Get started: To learn more, see the [Azure Active Directory developer's guide](#).

- **App Service Authentication:** When you choose App Service to host your app, you also get built-in authentication support for Azure AD, along with social identity providers— including Facebook, Google, Microsoft, and Twitter.

When to use: When you want to enable authentication in an App Service app by using Azure AD, social identity providers, or both.

Get started: To learn more about authentication in App Service, see [Authentication and authorization in Azure App Service](#).

To learn more about security best practices in Azure, see [Azure security best practices and patterns](#).

Monitoring

With your application up and running in Azure, you need to monitor performance, watch for issues, and see how customers are using your app. Azure provides several monitoring options.

- **Application Insights:** An Azure-hosted extensible analytics service that integrates with Visual Studio to monitor your live web applications. It gives you the data that you need to improve the performance and usability of your apps continuously. This improvement occurs whether you host your applications on Azure or not.

Get started: Follow the [Application Insights tutorial](#).

- **Azure Monitor:** A service that helps you to visualize, query, route, archive, and act on the metrics and logs that you generate with your Azure infrastructure and resources. Monitor is a single source for monitoring Azure resources and provides the data views that you see in the Azure portal.

DevOps integration

Whether it's provisioning VMs or publishing your web apps with continuous integration, Azure integrates with most of the popular DevOps tools. You can work with the tools that you already have and maximize your existing experience with support for tools like:

- Jenkins
- GitHub
- Puppet
- Chef
- TeamCity
- Ansible
- Azure DevOps



Azure regions

- Azure is a global cloud platform that is generally available in many regions around the world. When you provision a service, application, or VM in Azure, you're asked to select a region.
- This region represents a specific datacenter where your application runs or where your data is stored.
- These regions correspond to specific locations, which are published on the [Azure regions](#) page.

Choose the best region for your application and data

- One of the benefits of using Azure is that you can deploy your applications to various datacenters around the globe.
- The region that you choose can affect the performance of your application.
- For example, it's better to choose a region that's closer to most of your customers to reduce latency in network requests.

- You might also want to select your region to meet the legal requirements for distributing your app in certain countries/regions.
- It's always a best practice to store application data in the same datacenter or in a datacenter as near as possible to the datacenter that is hosting your application.

Multi-region apps

- Although unlikely, it's not impossible for an entire datacenter to go offline because of an event such as a natural disaster or Internet failure.
- It's a best practice to host vital business applications in more than one datacenter to provide maximum availability.
- Using multiple regions can also reduce latency for global users and provide additional opportunities for flexibility when updating applications.
- Some services, such as Virtual Machine and App Services, use [Azure Traffic Manager](#) to enable multi-region support with failover between regions to support high-availability enterprise applications.
- For an example, see [Azure reference architecture: Run a web application in multiple regions](#).

When to use: When you have enterprise and high-availability applications that benefit from failover and replication.

How do I manage my applications and projects?

Azure provides a rich set of experiences for you to create and manage your Azure resources, applications, and projects—both programmatically and in the [Azure portal](#).

Command-line interfaces and PowerShell

Azure provides two ways to manage your applications and services from the command line. You can use tools like Bash, Terminal, the command prompt, or your command-line tool of choice. Usually, you can do the same tasks from the command line as in the Azure portal—such as creating and configuring virtual machines, virtual networks, web apps, and other services.

- [Azure CLI](#): Lets you connect to an Azure subscription and program various tasks against Azure resources from the command line.
- [Azure PowerShell](#): Provides a set of modules with cmdlets that enable you to manage Azure resources by using Windows PowerShell.

Azure portal

The [Azure portal](#) is a web-based application. You can use the Azure portal to create, manage, and remove Azure resources and services. It includes:

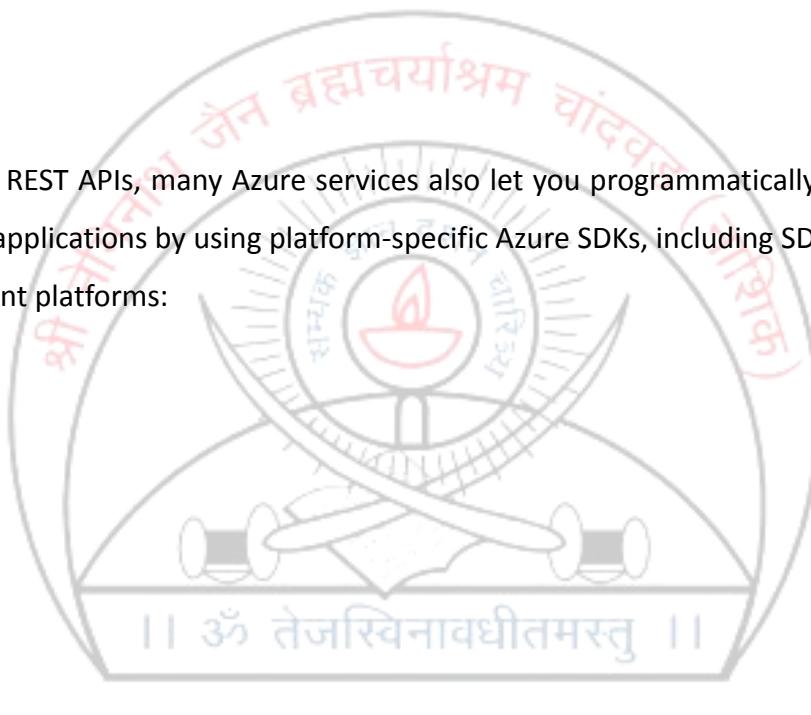
- A configurable dashboard
- Azure resource management tools
 - Access to subscription settings and billing information For more information, see the [Azure portal overview](#).

REST APIs

Azure is built on a set of REST APIs that support the Azure portal UI. Most of these REST APIs are also supported to let you programmatically provision and manage your Azure resources and applications from any Internet-enabled device. For the complete set of REST API documentation, see the [Azure REST SDK reference](#).

APIs

- Along with REST APIs, many Azure services also let you programmatically manage resources from your applications by using platform-specific Azure SDKs, including SDKs for the following development platforms:
- [.NET](#)
- [Node.js](#)
- [Java](#)
- [PHP](#)
- [Python](#)
- [Ruby](#)
- [Go](#)



Services such as [Mobile Apps](#) and [Azure Media Services](#) provide client-side SDKs to let you access services from web and mobile client apps.

Azure Resource Manager

- Running your app on Azure likely involves working with multiple Azure services. These services follow the same life cycle and can be thought of as a logical unit.
- For example, a web app might use Web Apps, SQL Database, Storage, Azure Cache for Redis, and Azure Content Delivery Network services.
- [Azure Resource Manager](#) lets you work with the resources in your application as a group. You can deploy, update, or delete all the resources in a single, coordinated operation.

Along with logically grouping and managing related resources, Azure Resource Manager includes deployment capabilities that let you customize the deployment and configuration of related resources. For example, you can use Resource Manager deploy and configure an application. This application can consist of multiple virtual machines, a load balancer, and a database in Azure SQL Database as a single unit.

- You develop these deployments by using an Azure Resource Manager template, which is a JSON-formatted document.
- Templates let you define a deployment and manage your applications by using declarative templates, rather than scripts. Your templates can work for different environments, such as testing, staging, and production. For example, you can use templates to add a button to a GitHub repo that deploys the code in the repo to a set of Azure services with a single click.

When to use: Use Resource Manager templates when you want a template-based deployment for your app that you can manage programmatically by using REST APIs, the Azure CLI, and Azure PowerShell.

Get started: To get started using templates, see [Authoring Azure Resource Manager templates](#).

Understanding accounts, subscriptions, and billing:

- As developers, we like to dive right into the code and try to get started as fast as possible with making our applications run.
- We certainly want to encourage you to start working in Azure as easily as possible. To help make it easy, Azure offers a [free trial](#).
- Some services even have a "Try it for free" functionality, like [Azure App Service](#), which doesn't require you to even create an account.
- As fun as it is to dive into coding and deploying your application to Azure, it's also important to take some time to understand how Azure works.
- Specifically, you should understand how it works from a standpoint of user accounts, subscriptions, and billing.

What is an Azure account?

- To create or work with an Azure subscription, you must have an Azure account.
- An Azure account is simply an identity in Azure AD or in a directory, such as a work or school organization, that Azure AD trusts.

- If you don't belong to such an organization, you can always create a subscription by using your Microsoft Account, which is trusted by Azure AD.
- To learn more about integrating on-premises Windows Server Active Directory with Azure AD,
- Every Azure subscription has a trust relationship with an Azure AD instance. This means that it trusts that directory to authenticate users, services, and devices.
- Multiple subscriptions can trust the same directory, but a subscription trusts only one directory. To learn more, see [How Azure subscriptions are associated with Azure Active Directory](#).
- As well as defining individual Azure account identities, also called *users*, you can define *groups* in Azure AD.
- Creating user groups is a good way to manage access to resources in a subscription by using role-based access control (RBAC).
- To learn how to create groups, see [Create a group in Azure Active Directory preview](#). You can also create and manage groups by [using PowerShell](#).

Manage your subscriptions

- A subscription is a logical grouping of Azure services that is linked to an Azure account. A single Azure account can contain multiple subscriptions.
- Billing for Azure services is done on a per-subscription basis. For a list of the available subscription offers by type, see [Microsoft Azure Offer Details](#).
- Azure subscriptions have an Account Administrator who has full control over the subscription.
- They also have a Service Administrator who has control over all services in the subscription.
- For information about classic subscription administrators, see [Add or change Azure subscription administrators](#). Individual accounts can be granted detailed control of Azure resources using [Azure role-based access control \(Azure RBAC\)](#).

Resource groups

- When you provision new Azure services, you do so in a given subscription. Individual Azure services, which are also called resources, are created in the context of a resource group. Resource groups make it easier to deploy and manage your application's resources.
- A resource group should contain all the resources for your application that you want to work with as a unit.
- You can move resources between resource groups and even to different subscriptions.

The Azure Resource Explorer is a great tool for visualizing the resources that you've already created in your subscription. To learn more, see [Use Azure Resource Explorer to view and modify resources](#).

Grant access to resources

When you allow access to Azure resources, it's always a best practice to provide users with the least privilege that's required to do a given task.

- **Azure role-based access control (Azure RBAC):** In Azure, you can grant access to user accounts (principals) at a specified scope: subscription, resource group, or individual resources. Azure RBAC lets you deploy resources into a resource group and grant permissions to a specific user or group. It also lets you limit access to only the resources that belong to the target resource group. You can also grant access to a single resource, such as a virtual machine or virtual network. To grant access, you assign a role to the user, group, or service principal. There are many predefined roles, and you can also define your own custom roles.

When to use: When you need fine-grained access management for users and groups or when you need to make a user an owner of a subscription.

- **Service principal objects:** Along with providing access to user principals and groups, you can grant the same access to a service principal.

When to use: When you're programmatically managing Azure resources or granting access for applications. For more information, see [Create Active Directory application and service principal](#).

Tags

Azure Resource Manager lets you assign custom tags to individual resources. Tags, which are key-value pairs, can be helpful when you need to organize resources for billing or monitoring.

Tags provide you a way to track resources across multiple resource groups. You can assign tags the following ways:

- In the portal
- In the Azure Resource Manager template
- Using the REST API
- Using the Azure CLI
- Using PowerShell

You can assign multiple tags to each resource. To learn more, see [Using tags to organize your Azure resources](#).

Billing

In the move from on-premises computing to cloud-hosted services, tracking and estimating service usage and related costs are significant concerns. It's important to estimate what new resources cost to run on a monthly basis. You can also project how the billing looks for a given month based on the current spending.

Get resource usage data

Azure provides a set of Billing REST APIs that give access to resource consumption and metadata information for Azure subscriptions. These Billing APIs give you the ability to better predict and manage Azure costs. You can track and analyze spending in hourly increments and create spending alerts. You can also predict future billing based on current usage trends.

Get started: To learn more about using the Billing APIs, see [Azure consumption API overview](#)

Predict future costs

Although it's challenging to estimate costs ahead of time, Azure has tools that can help. It has a [pricing calculator](#) to help estimate the cost of deployed resources. You can also use the Billing resources in the portal and the Billing REST APIs to estimate future costs, based on current consumption.

Create a static HTML web app in Azure

[Azure App Service](#) provides a highly scalable, self-patching web hosting service. This quickstart shows how to deploy a basic HTML+CSS site to Azure App Service. You'll complete this quickstart in [Cloud Shell](#), but you can also run these commands locally with [Azure CLI](#).

If you don't have an [Azure subscription](#), create an [Azure free account](#) before you begin.

Use Azure Cloud Shell

Azure hosts Azure Cloud Shell, an interactive shell environment that you can use through your browser. You can use either Bash or PowerShell with Cloud Shell to work with Azure services. You can use the Cloud Shell preinstalled commands to run the code in this article without having to install anything on your local environment.

To start Azure Cloud Shell:

Select **Try It** in the upper-right corner of a code block. Selecting **Try It** doesn't automatically copy the code to Cloud Shell.

Go to <https://shell.azure.com>, or select the **Launch Cloud Shell** button to open Cloud Shell in your browser.

Select the **Cloud Shell** button on the menu bar at the upper right in the [Azure portal](#).

To run the code in this article in Azure Cloud Shell:

1. Start Cloud Shell.
2. Select the **Copy** button on a code block to copy the code.
3. Paste the code into the Cloud Shell session by selecting **Ctrl+Shift+V** on Windows and Linux or by selecting **Cmd+Shift+V** on macOS.
4. Select **Enter** to run the code.

Download the sample

In the Cloud Shell, create a quickstart directory and then change to it.

mkdir quickstart

cd \$HOME/quickstart

Next, run the following command to clone the sample app repository to your quickstart directory.

git clone <https://github.com/Azure-Samples/html-docs-hello-world.git>

Create a web app

Change to the directory that contains the sample code and run the az webapp up command. In the following example, replace <app_name> with a unique app name. Static content is indicated by the --html flag.

Bash

cd html-docs-hello-world

az webapp up --location westeurope --name <app_name> --html

The az webapp up command does the following actions:

- Create a default resource group.

- Create a default app service plan.
- Create an app with the specified name.
- [Zip deploy](#) files from the current working directory to the web app.

This command may take a few minutes to run. While running, it displays information similar to the following example:

Output

```
{
  "app_url": "https://<app_name>.azurewebsites.net", "location": "westeurope",
  "name": "<app_name>",
  "os": "Windows",
  "resourcegroup": "appsvc_rg_Windows_westeurope", "serverfarm": "appsvc_asp_Windows_westeurope", "sku": "FREE",
  "src_path": "/home/<username>/quickstart/html-docs-hello-world ", &lt; JSON data removed for brevity. &gt;
}
```

Make a note of the resourceGroup value. You need it for the [clean up resources](#) section.

Browse to the app

In a browser, go to the app URL: http://<app_name>.azurewebsites.net. The page is running as an Azure App Service web app.

ESTD - 1928

Congratulations! You've deployed your first HTML app to App Service.

Update and redeploy the app

- In the Cloud Shell, type nano index.html to open the nano text editor. In the <h1> heading tag, change "Azure App Service - Sample Static HTML Site" to "Azure App Service", as shown below.
- Save your changes and exit nano. Use the command ^O to save and ^X to exit. You'll now redeploy the app with the same az webapp up command.

Bash

```
az webapp up --location westeurope --name <app_name> --html
```

Once deployment has completed, switch back to the browser window that opened in the **Browse to the app** step, and refresh the page.

Manage your new Azure app

To manage the web app you created, in the [Azure portal](#), search for and select **App Services**.

On the **App Services** page, select the name of your Azure app.

You see your web app's Overview page. Here, you can perform basic management tasks like browse, stop, start, restart, and delete.

The left menu provides different pages for configuring your app.

Clean up resources

In the preceding steps, you created Azure resources in a resource group. If you don't expect to need these resources in the future, delete the resource group by running the following command in the Cloud Shell. Remember that the resource group name was automatically generated for you in the [create a web app](#) step.

Bash

```
az group delete --name appsvc_rg_Windows_westeurope This command may take a minute to run.
```

6.Frequently Asked Questions:

- i) What is Azure?
- ii) What is Azure Virtual Machine?
- iii) What is Azure App Service?
- iv) What is Azure Service Fabric?
- v) What is Azure Spring Cloud?
- vi) What is difference between Microsoft Azur, Google Cloud, Amazon EC2

7.Conclusion:

Successfully studied Microsoft Azure Platform and created a static HTML web app in Azure

SNJB's Late Sau. K. B. Jain College of Engineering, Chandwad

Department of Computer Engineering

Course Name:Laboratory Practice II(310258):Cloud Computing

Class:Third Year (TE) Div A/ Div B

Batch:T1/T2/T3/T4

Name:

Roll No:

Assignment No: 8

Answers (A) – 5M	Coding Efficiency (C) – 5M	Viva (V) – 5M	Timely Completion (T) – 5M	Total(20M)	Sign

Date of Performance:..... **Date of Completion:**.....

1. Title of Assignment:

Installation and configure Google App Engine.

2. Objective:

1. Install and Configure Google App Engine
2. Create a simple code in python on Github and execute on Google App Engine

3. Outcome: Use tools and techniques in the area of Cloud Computing

4. Software and Hardware Requirement:

Software Requirement: 1.Console.cloud.google.com 2. Login to your Gmail account

Hardware Requirement: Internet Connection, PC with Min. 2GB RAM, Core i5 Processor

5.Relevant Theory :

Google App Engine :

Google App Engine (often referred to as GAE or simply App Engine) is a cloud computing platform as a service for developing and hosting web applications in Google-managed data centers. Applications are sandboxed and run across multiple servers. App Engine offers automatic scaling for web applications—as the number of requests increases for an application, App Engine automatically allocates more resources for the web application to handle the additional demand.

Google App Engine primarily supports Go, PHP, Java, Python, Node.js, .NET, and Ruby

applications, although it can also support other languages via "custom runtimes". The service is free up to a certain level of consumed resources and only in standard environment but not in flexible environment. Fees are charged for additional storage, bandwidth, or instance hours required by the application. It was first released as a preview version in April 2008 and came out of preview in September 2011.

Runtimes and framework

Python web frameworks that run on Google App Engine include Django, CherryPy, Pyramid, Flask, web2py and webapp2, as well as a custom Google- written webapp framework and several others designed specifically for the platform that emerged since the release. Any Python framework that supports the WSGI using the CGI adapter can be used to create an application; the framework can be uploaded with the developed application. Third-party libraries written in pure Python may also be uploaded.

Google App Engine supports many Java standards and frameworks. Core to this is the servlet 2.5 technology using the open-source Jetty Web Server, along with accompanying technologies such as JSP. JavaServer Faces operates with some workarounds. A newer release of App Engine Standard Java in Beta supports Java8, Servlet 3.1 and Jetty9.

Though the integrated database, Google Cloud Datastore, may be unfamiliar to programmers, it is accessed and supported with JPA, JDO, and by the simple low-level API. There are several alternative libraries and frameworks you can use to model and map the data to the database such as Objectify, Slim3 and Jello framework.

The Spring Framework works with GAE. However, the Spring Security module (if used) requires workarounds. Apache Struts 1 is supported, and Struts 2 runs with workarounds.

The Django web framework and applications running on it can be used on App Engine with modification. Django-nonrel aims to allow Django to work with non-relational databases and the project includes support for App Engine.

Reliability and support

All billed App Engine applications have a 99.95% uptime SLA.

App Engine is designed in such a way that it can sustain multiple datacenter outages without any downtime. This resilience to downtime is shown by the statistic that the High Replication Datastore saw 0% downtime over a period of a year.

Paid support from Google engineers is offered as part of Premier Accounts.

Differences with other application hosting

Compared to other scalable hosting services such as Amazon EC2, App Engine provides more infrastructure to make it easy to write scalable applications, but can only run a limited range of applications designed for that infrastructure.

App Engine's infrastructure removes many of the system administration and development challenges of building applications to scale to hundreds of requests per second and beyond. Google handles deploying code to a cluster, monitoring, failover, and launching application instances as necessary. While other services let users install and configure nearly any *NIX compatible software, App Engine requires developers to use only its supported languages, APIs, and frameworks. Current APIs allow storing and retrieving data from the document-oriented Google Cloud Datastore database;

making HTTP requests; sending e-mail; manipulating images; and caching. Google Cloud SQL can be used for App Engine applications requiring a relational MySQL compatible database backend.

Per-day and per-minute quotas restrict bandwidth and CPU use, number of requests served, number of concurrent requests, and calls to the various APIs, and individual requests are terminated if they take more than 60 seconds or return more than 32MB of data.

Differences between SQL and GQL

Google App Engine's integrated Google Cloud Datastore database has a SQL-like syntax called "GQL" (Google Query Language). GQL does not support the Join statement. Instead, one-to-many and many-to-many relationships can be accomplished using ReferenceProperty() .

Google Firestore is the successor to Google Cloud Datastore and replaces GQL with a document-based query method that treats stored objects as collections of documents.

Portability concerns

Developers worry that the applications will not be portable from App Engine and fear being locked into the technology. In response, there are a number of projects to create open-source back-ends for the various proprietary/closed APIs of app engine, especially the datastore. AppScale, CapeDwarf and TyphoonAE are a few of the open source efforts.

AppScale automatically deploys and scales unmodified Google App Engine applications over popular public and private cloud systems and on-premises clusters. AppScale can run Python, Java, PHP, and Go applications on EC2, Google Compute Engine, Softlayer, Azure and other cloud vendors.

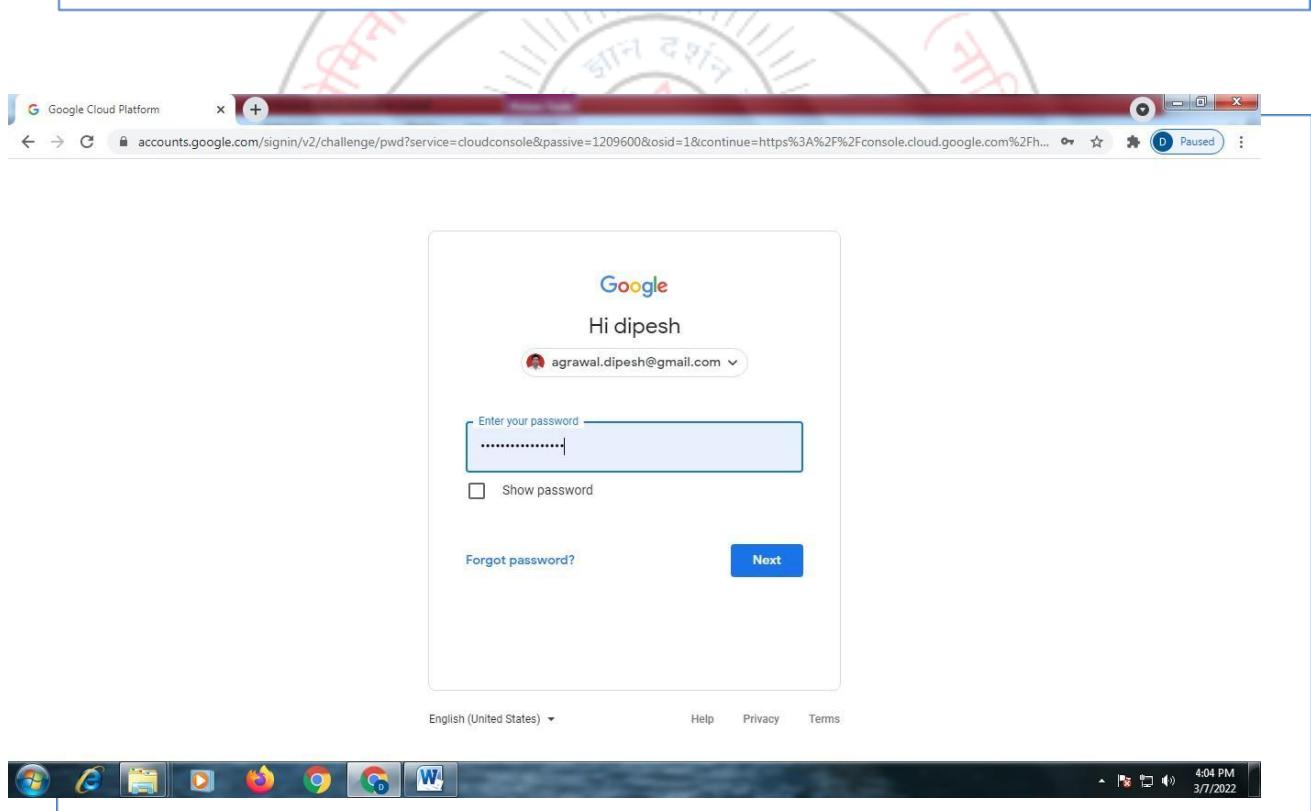
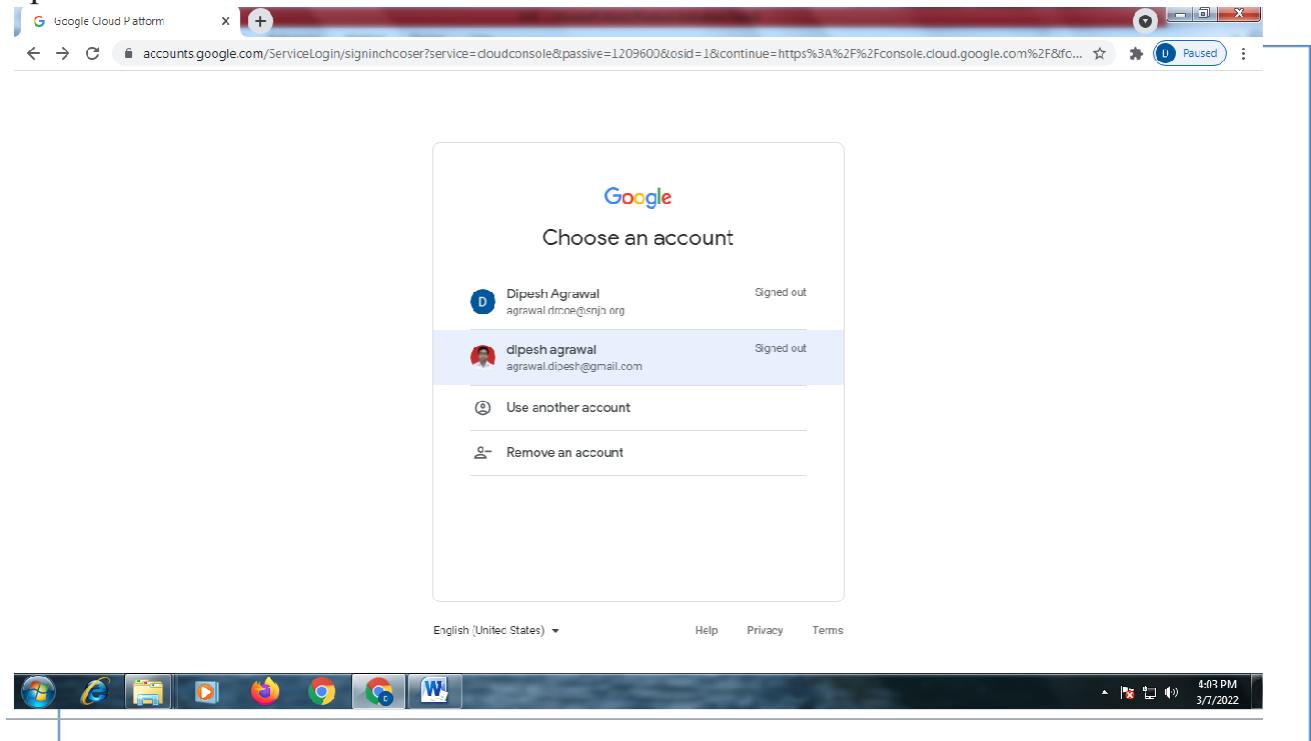
TyphoonAE can run Python App Engine applications on any cloud that support linux machines.

Web2py web framework offers migration between SQL Databases and Google App Engine, however it doesn't support several App Engine-specific features such as transactions and namespaces.

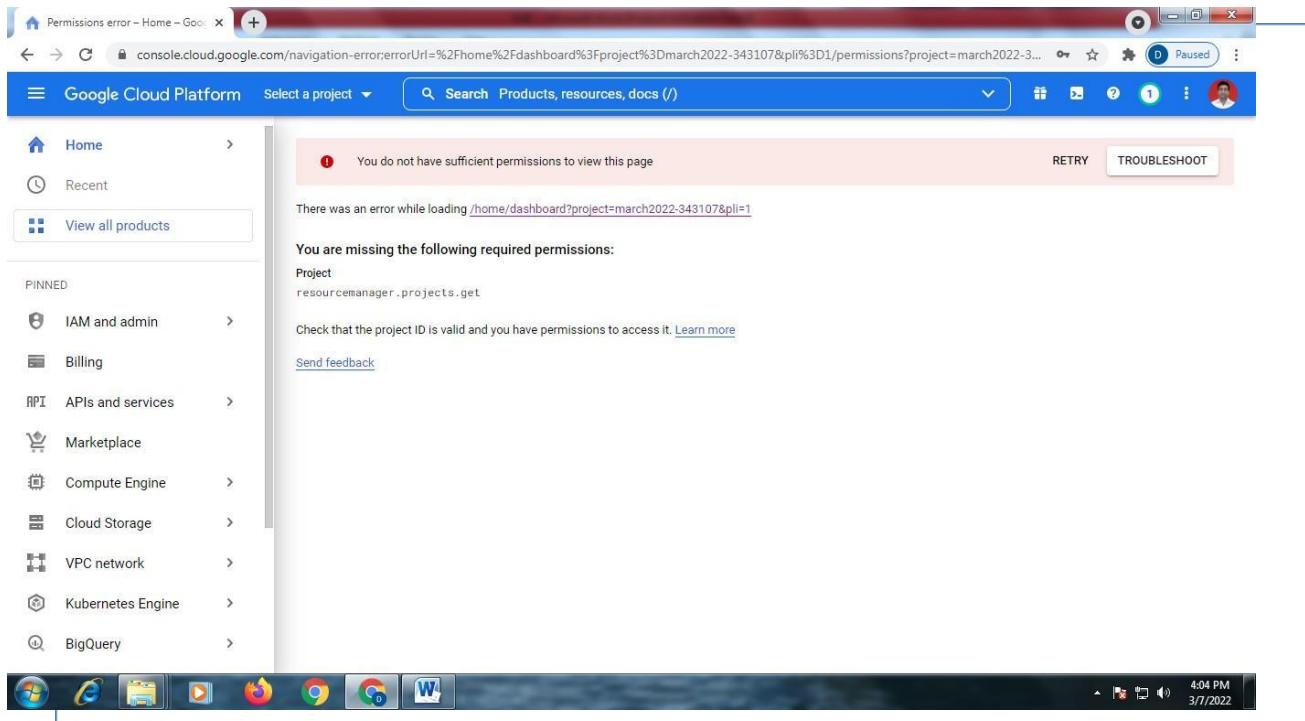
Kubernetes is an open-source job control system invented by Google to abstract away the infrastructure so that open-source (e.g. Docker) containerized applications can run on many types of infrastructure, such as Amazon Web Services, Microsoft Azure, and others. This is one of Google's answers to the portability concern.

ESTD - 1928

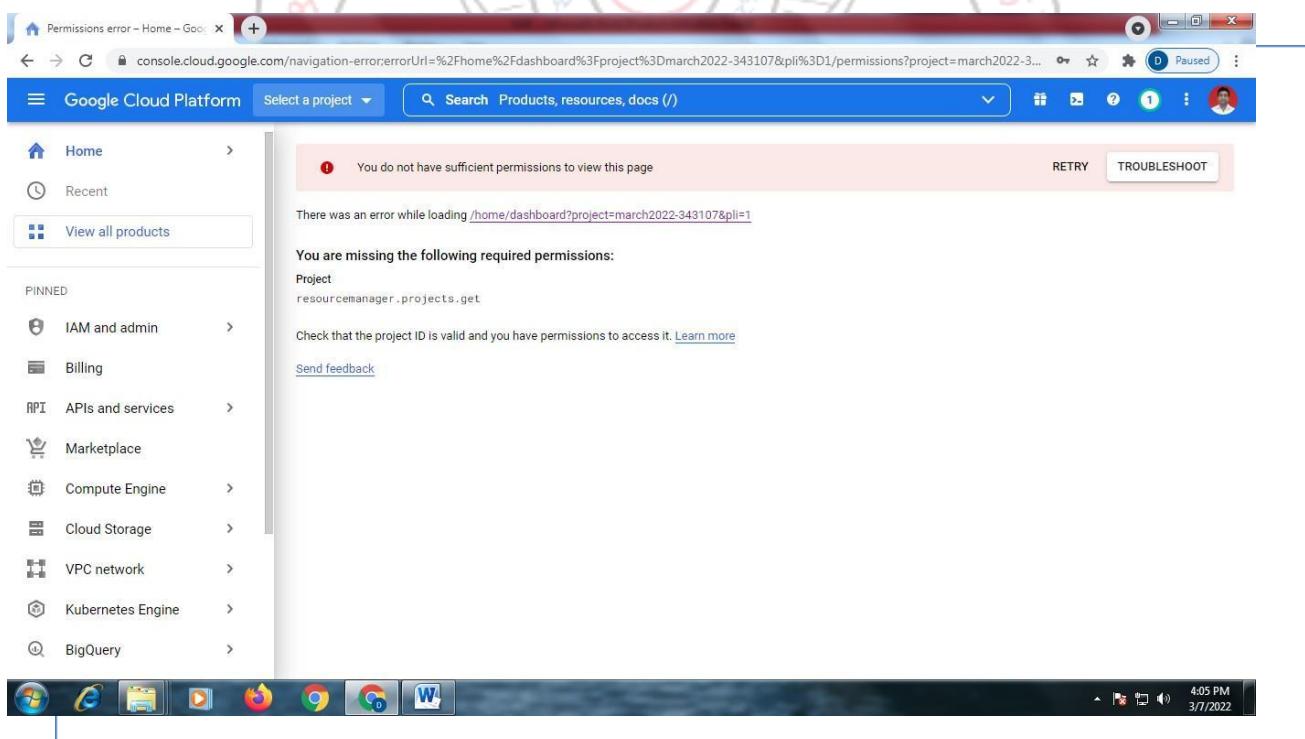
Steps □

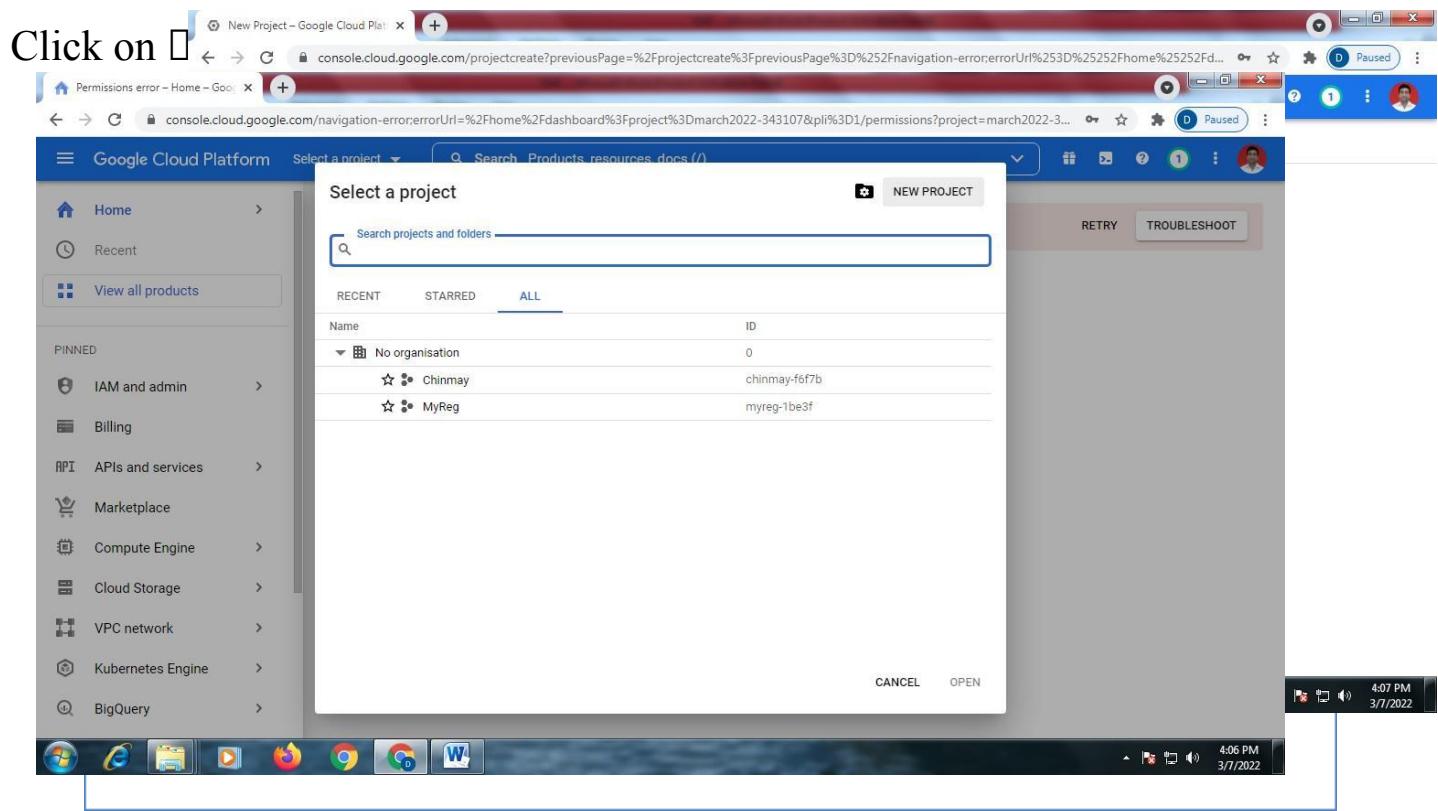


Following Screen Will Appear

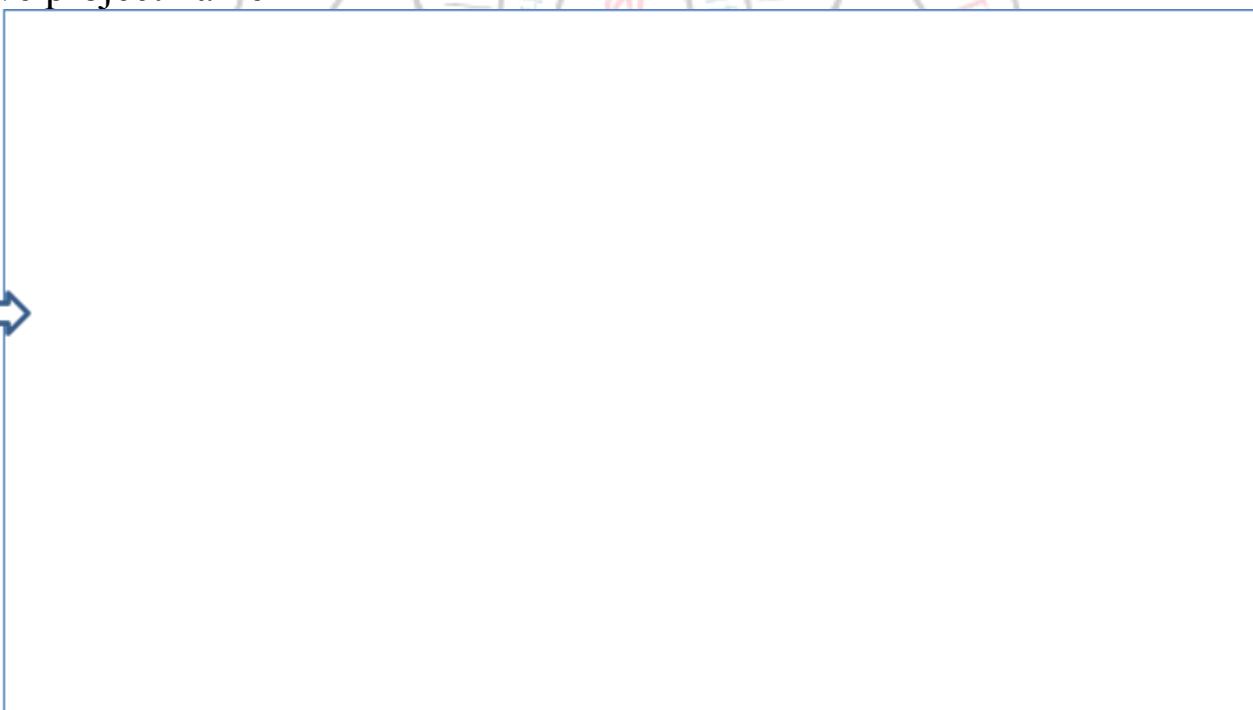


Click on Select new project

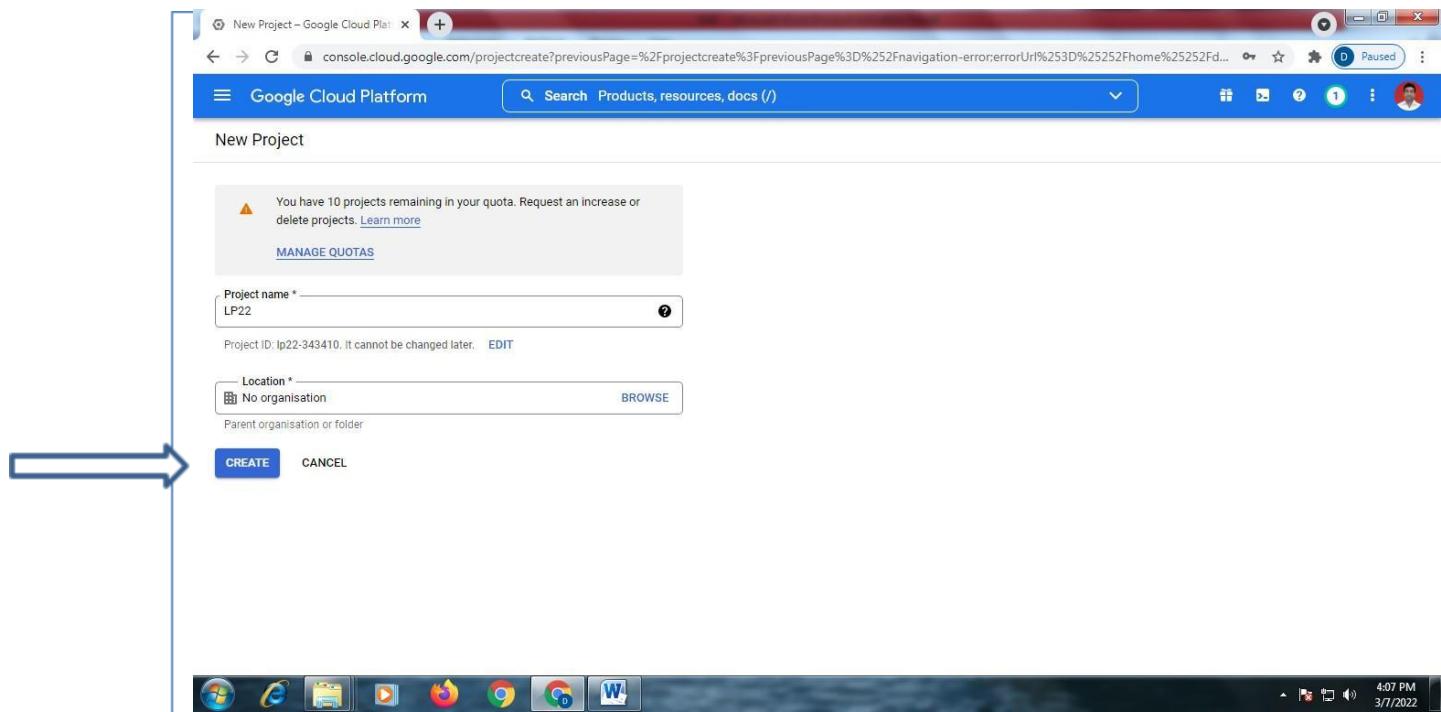




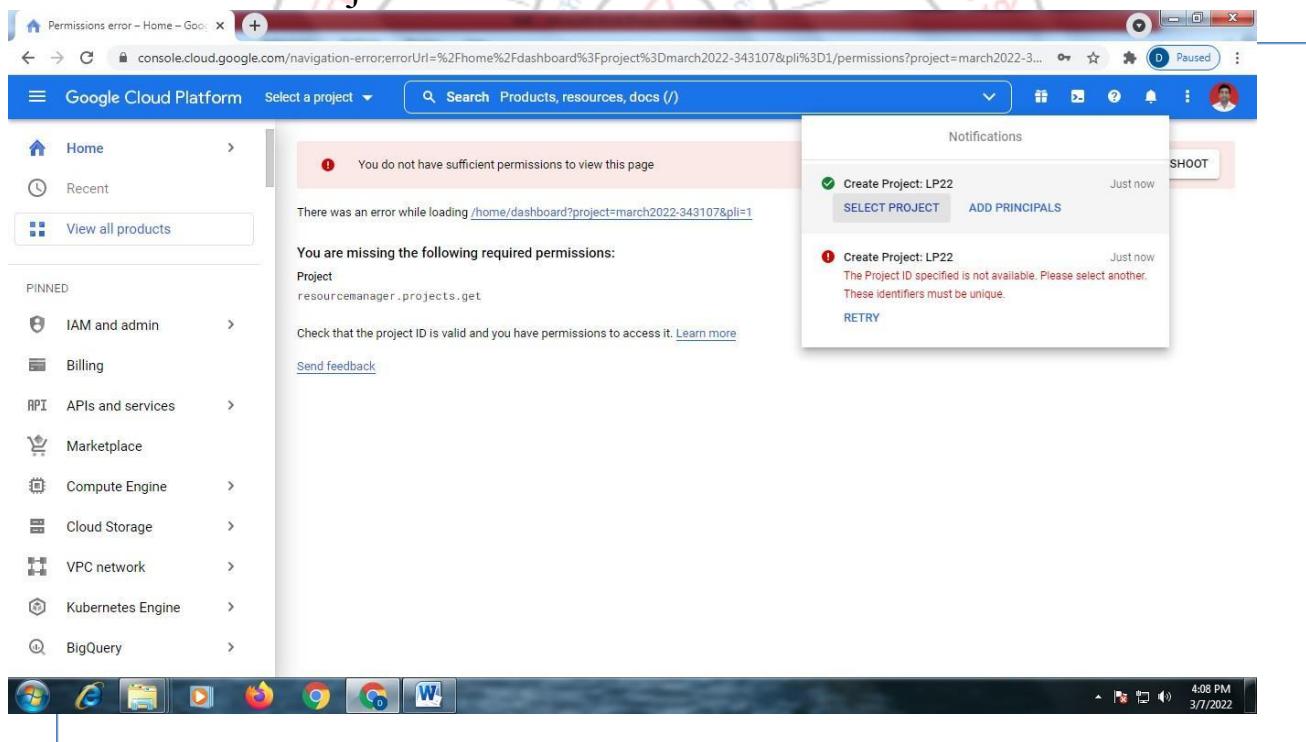
Give project name



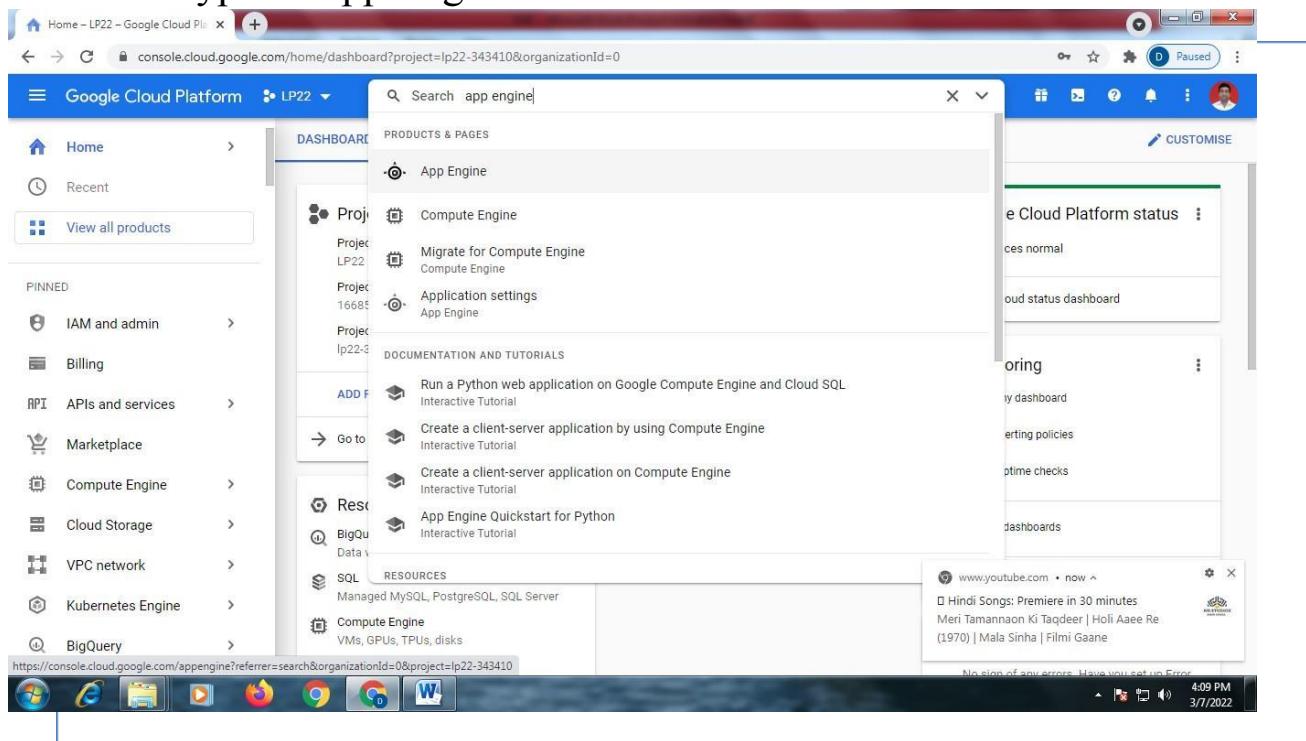
Click on create



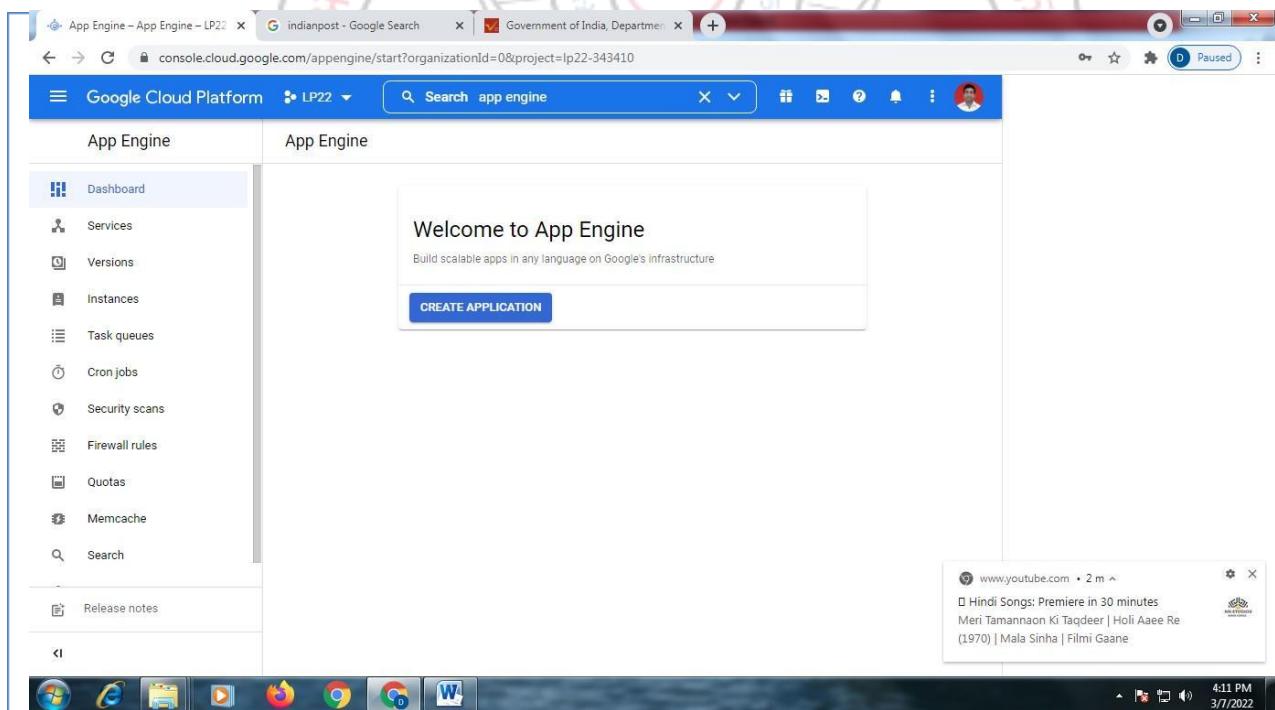
Click on Select Project

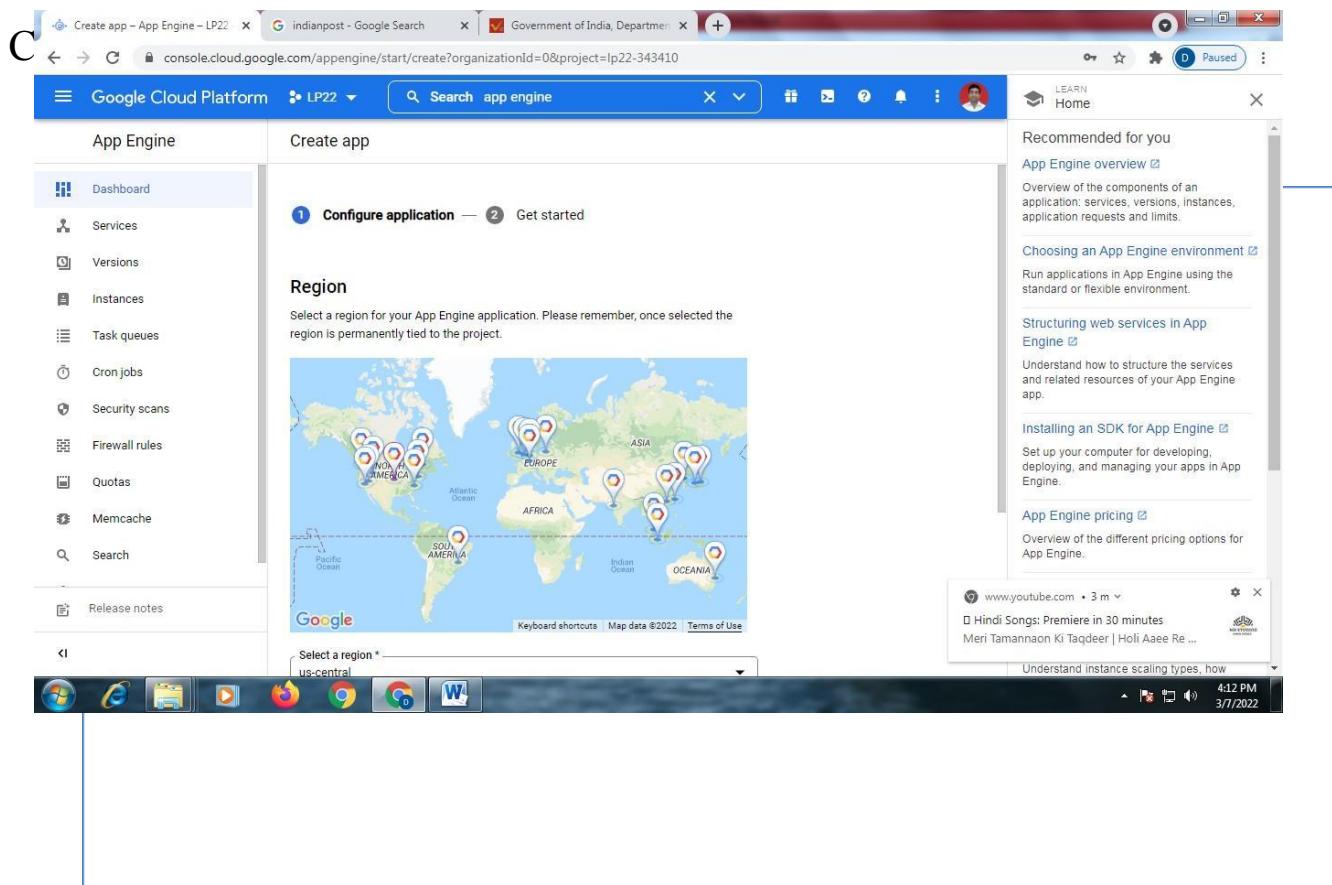


In search type —App Engine|

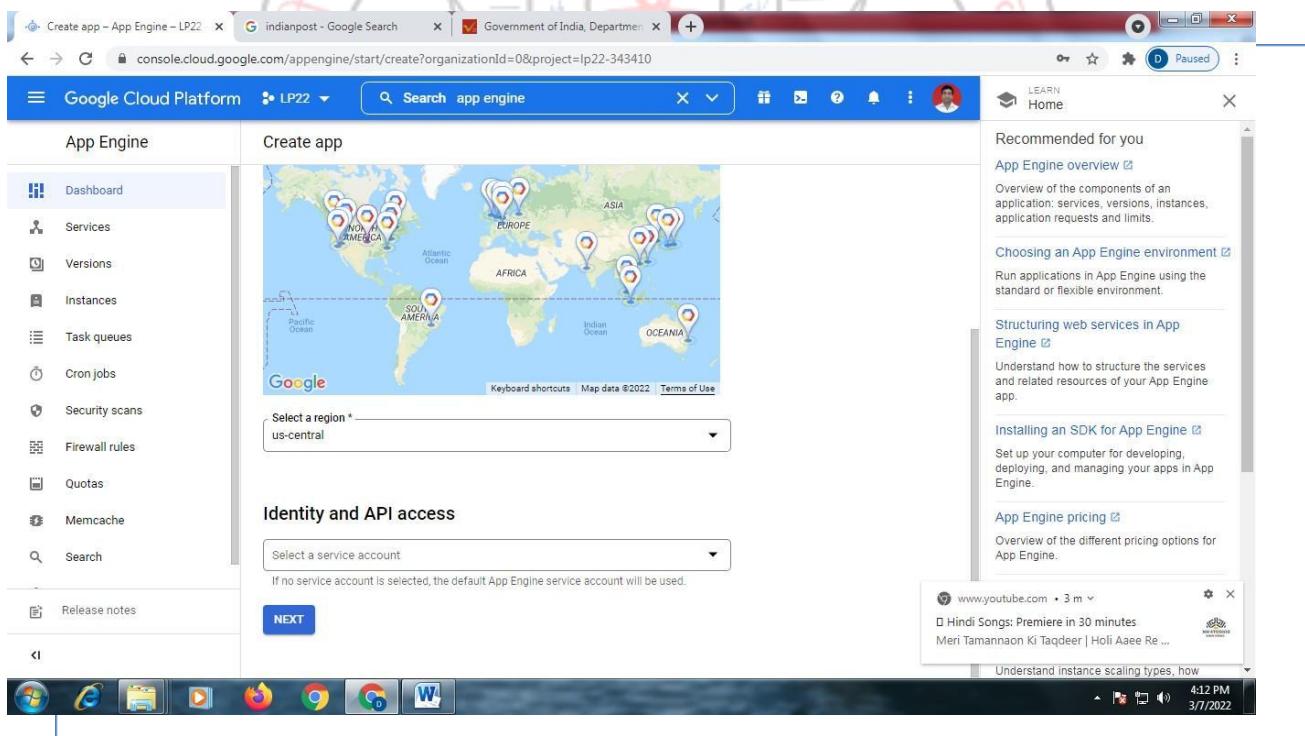


Click on ——App Engine| Following screen will appear



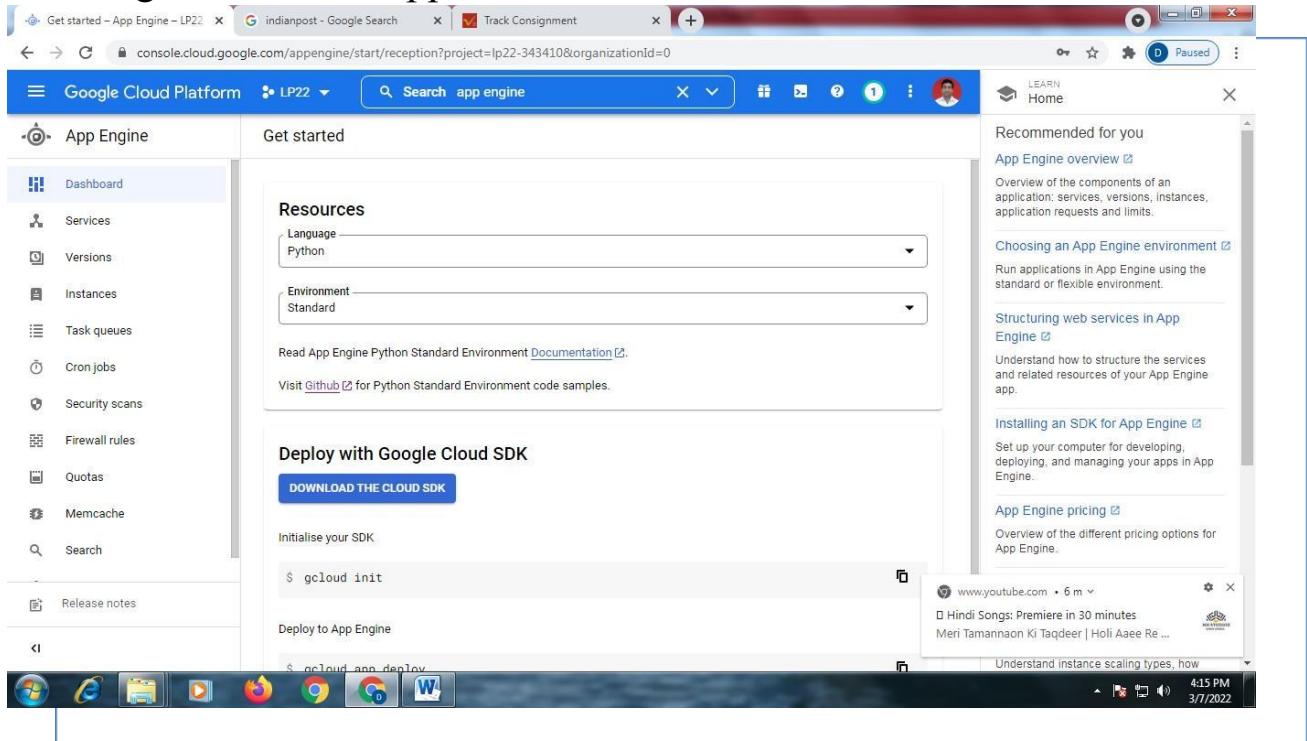


Scroll Down

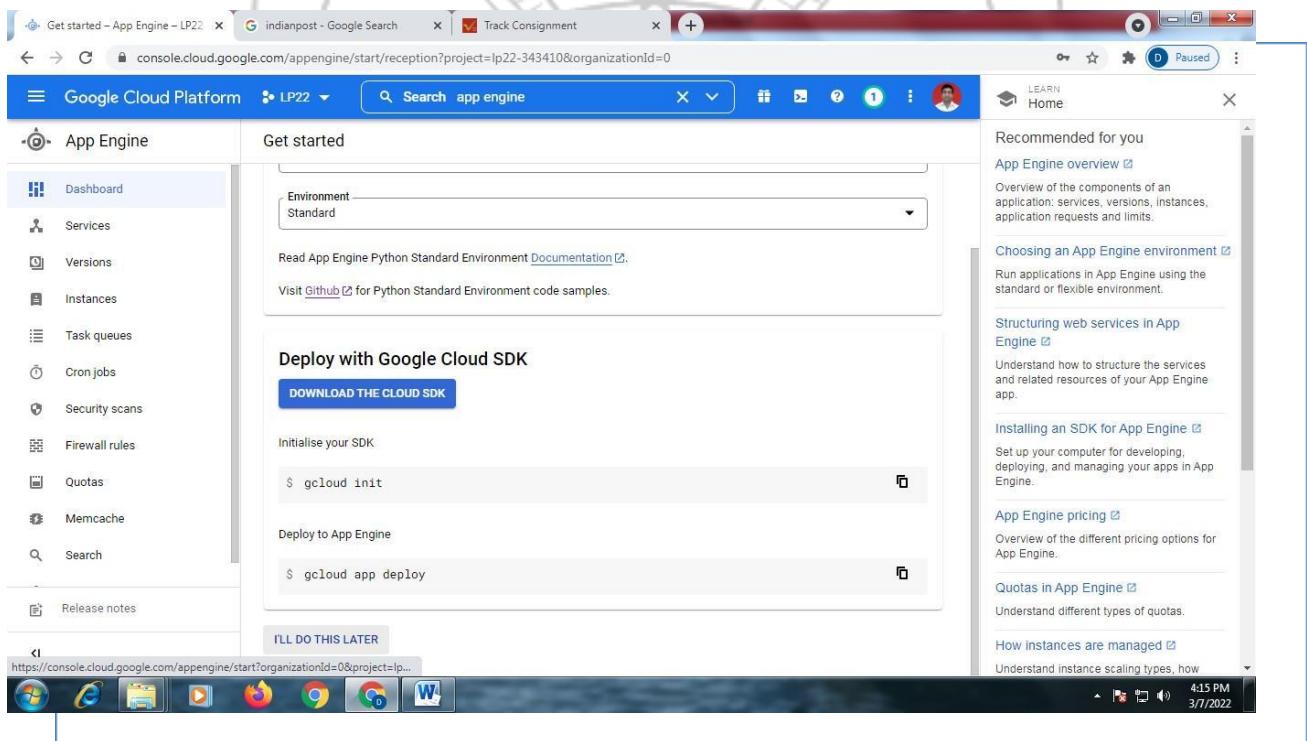


Click on —Next||

Following Screen Will Appear



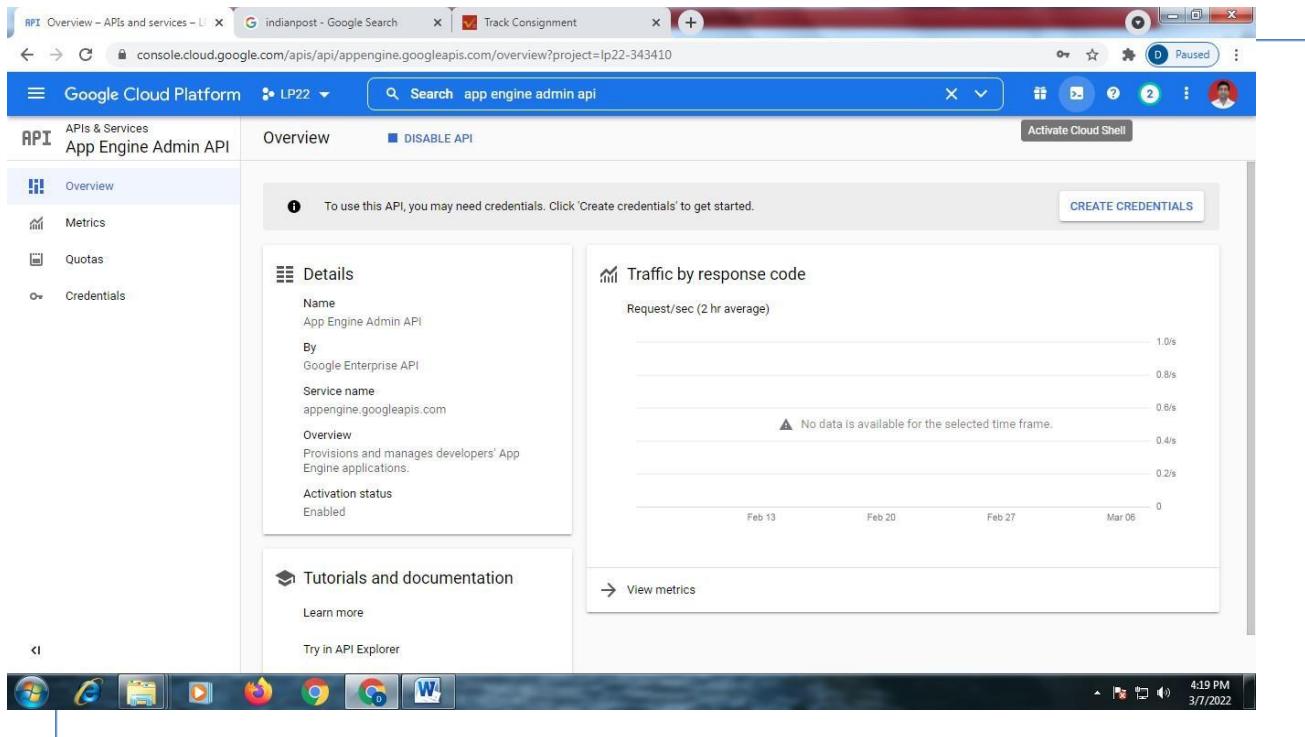
Scroll Down Click on — I will do it Later||



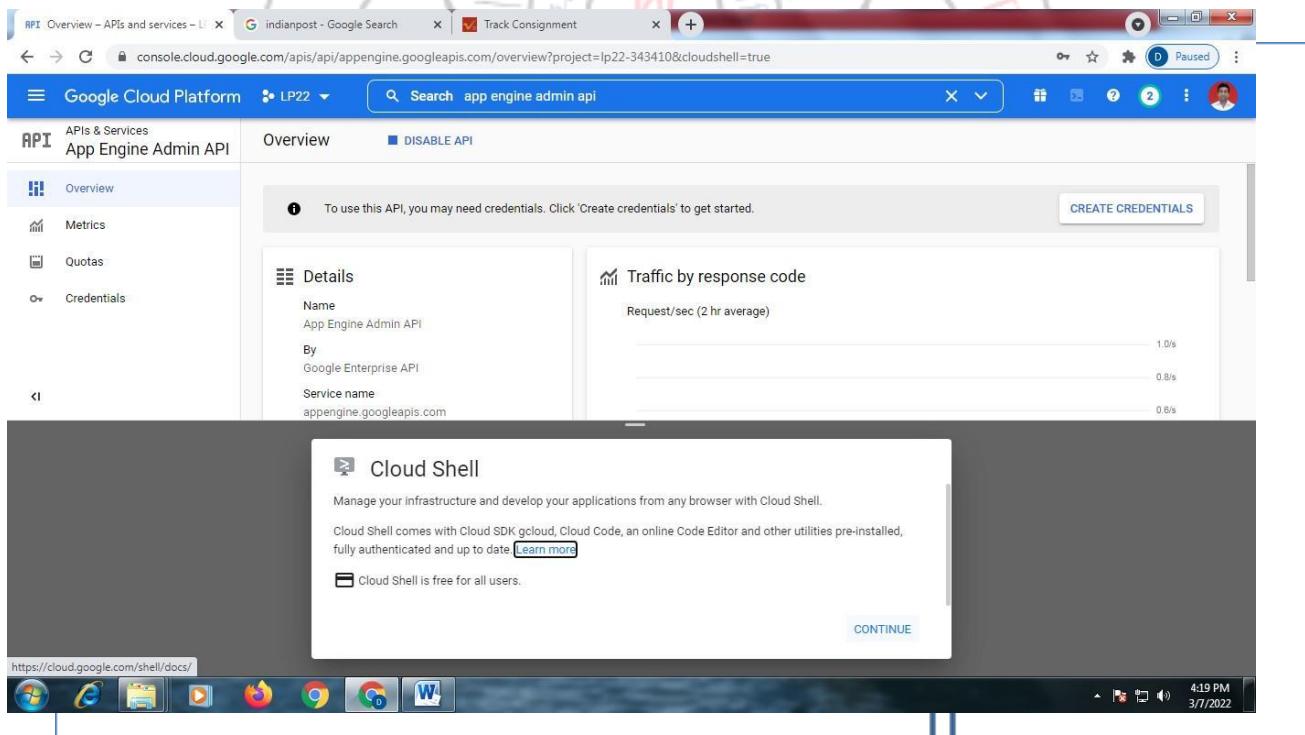
In search box, type —App Engine Admin API||

Click on ||App Engine Admin API|| Click on —Enable||

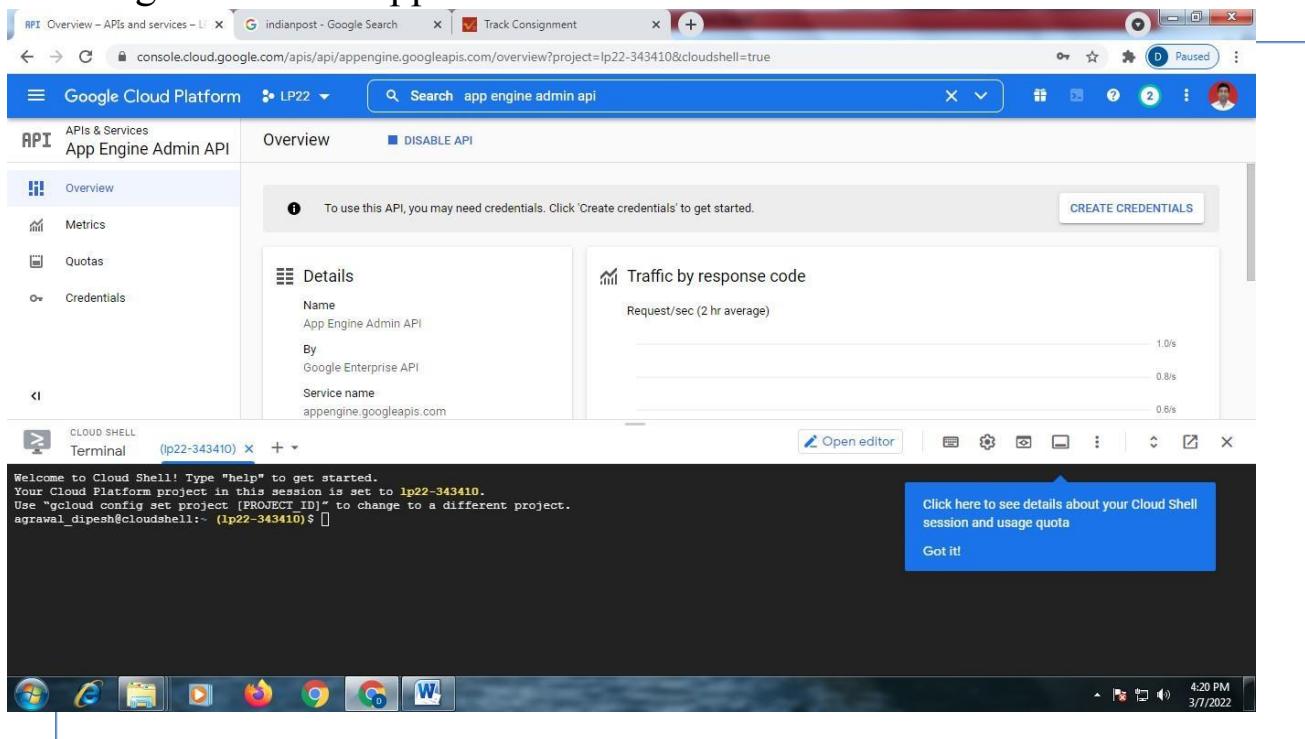
Click on —Activate Cloud Shell||



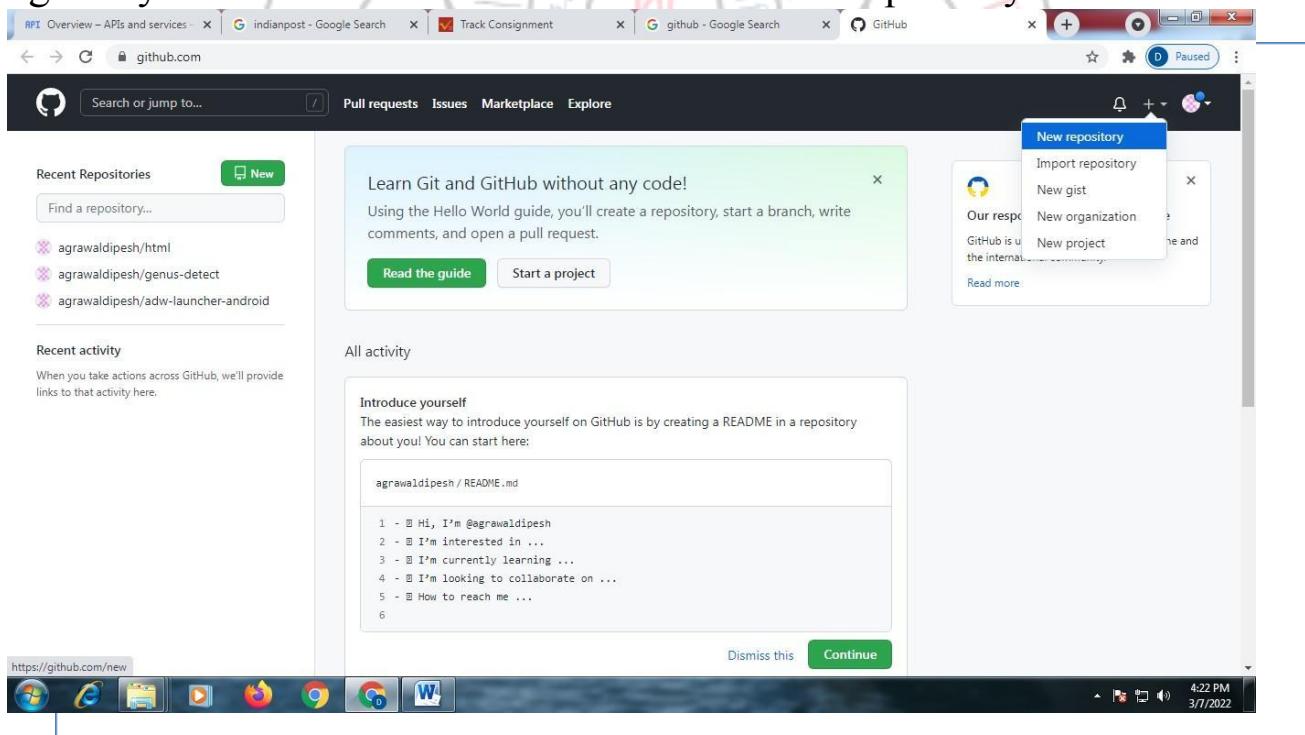
Click on —Continue||



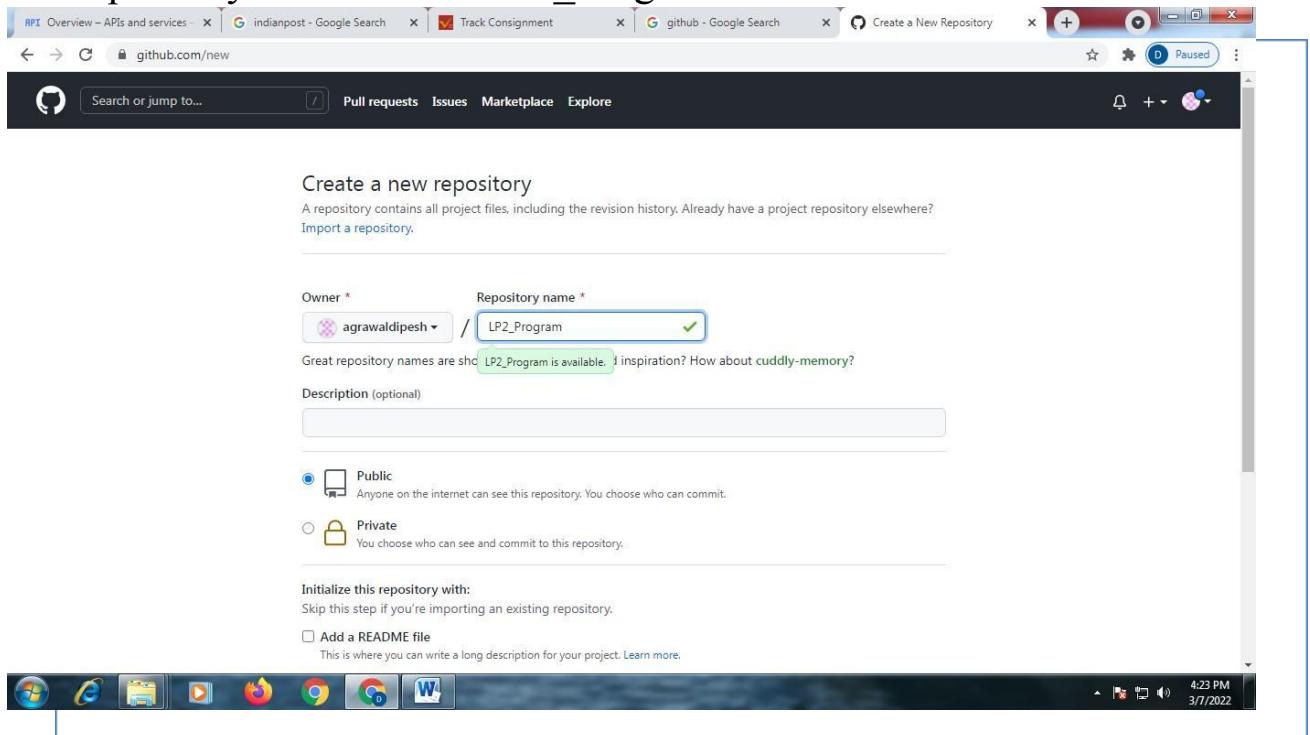
Following Screen will appear



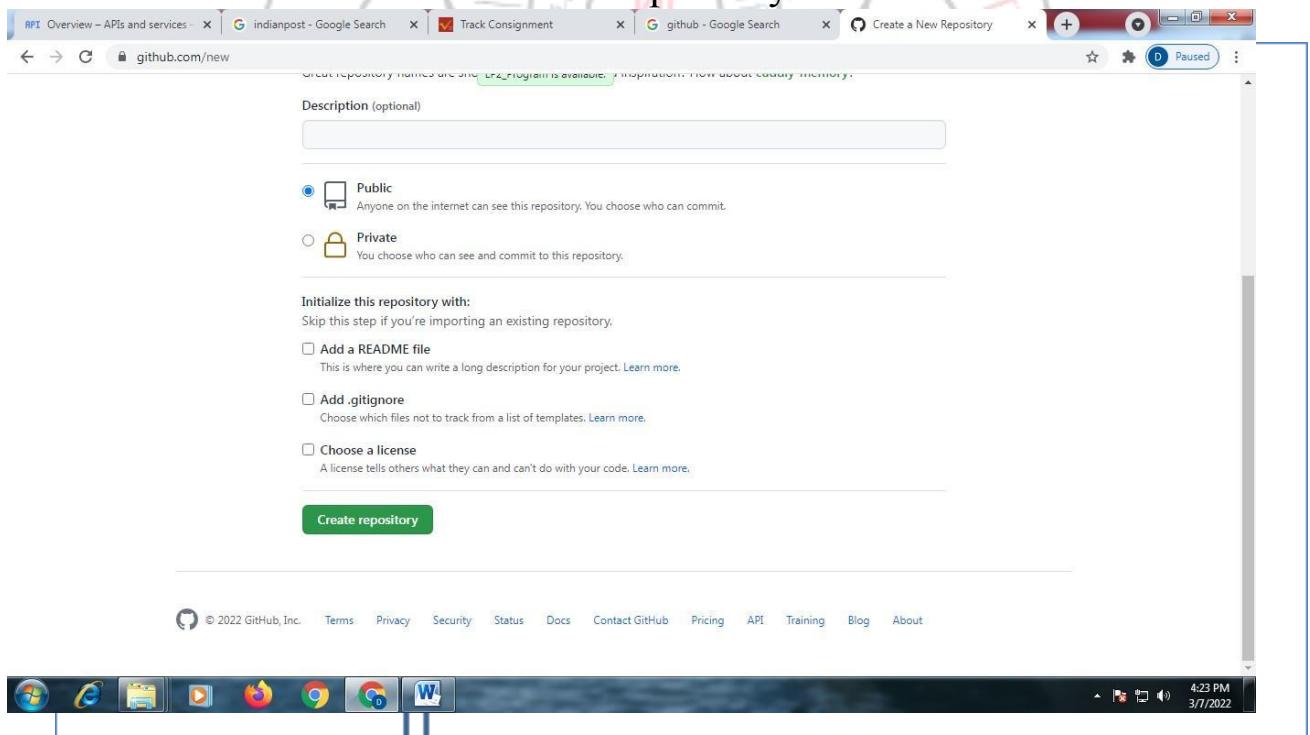
Login to your Github Account □ Click on —New Repository||



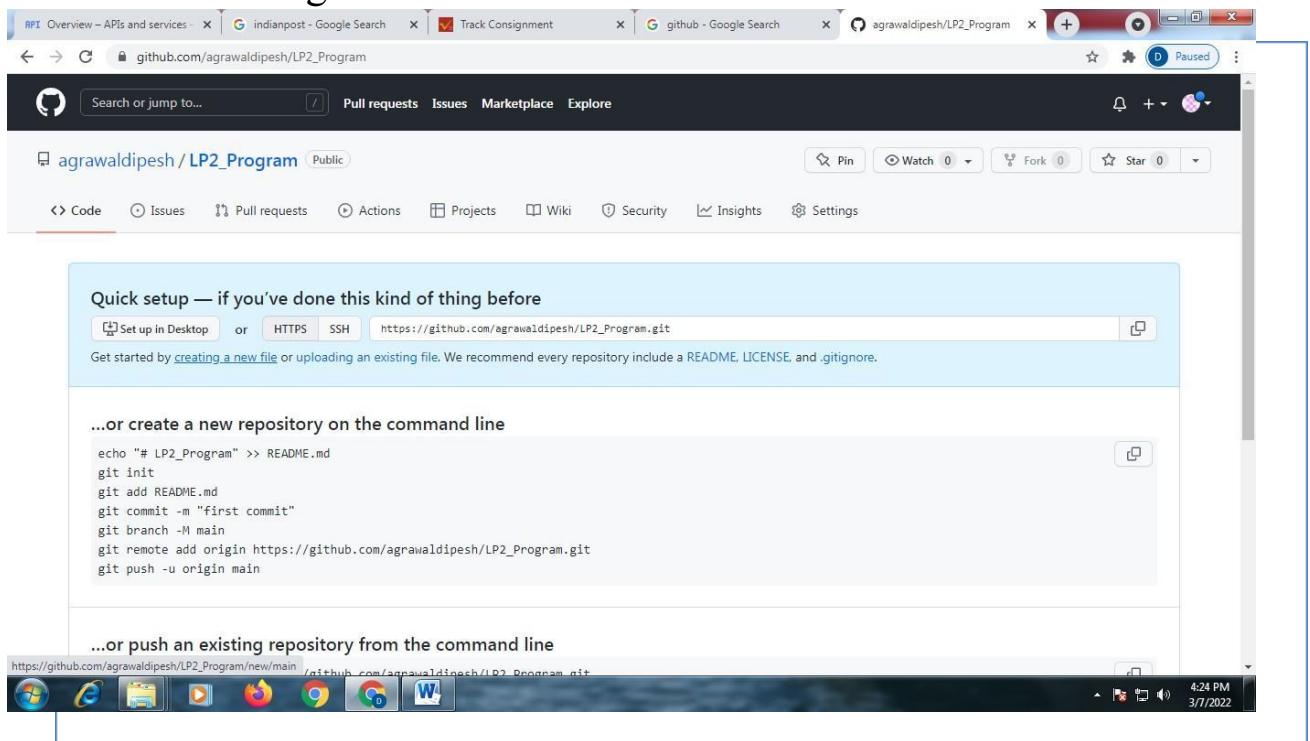
Give repository name as —LP2_Program||



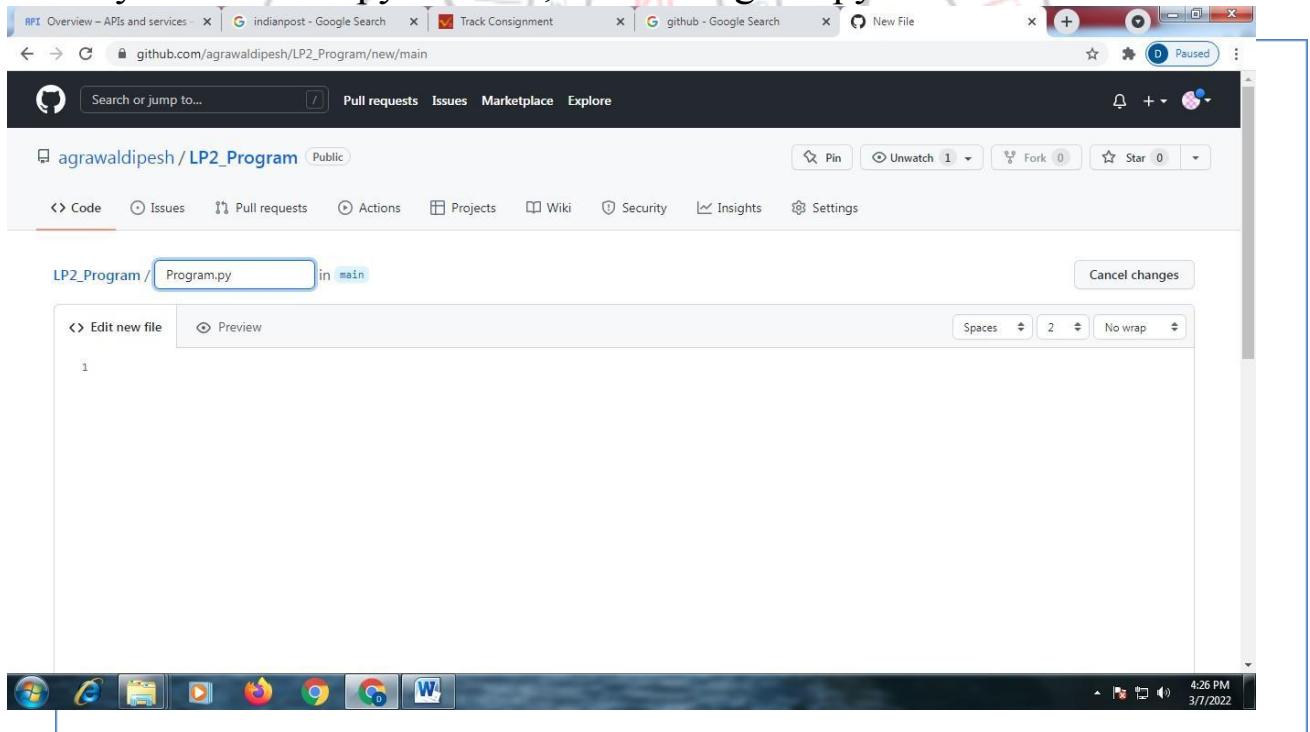
Scroll Down and Click on —Create Repository||



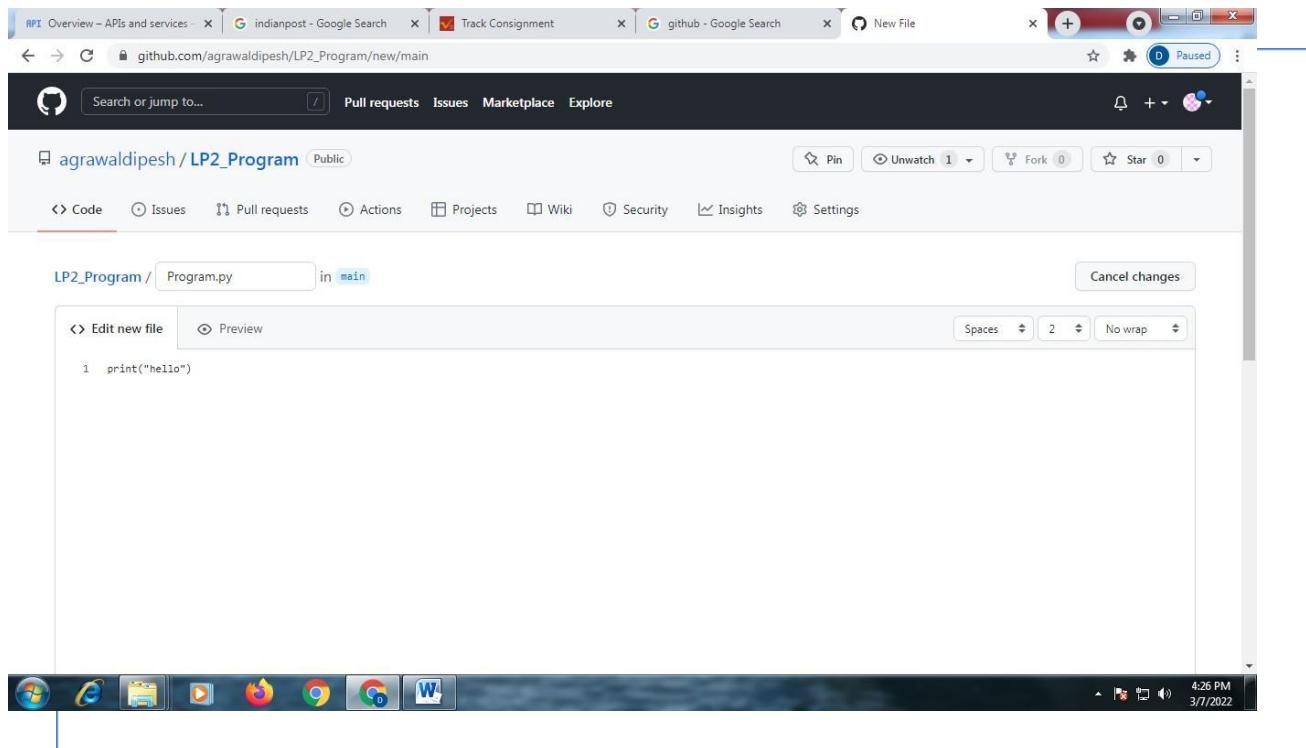
Click on —Creating a new File||



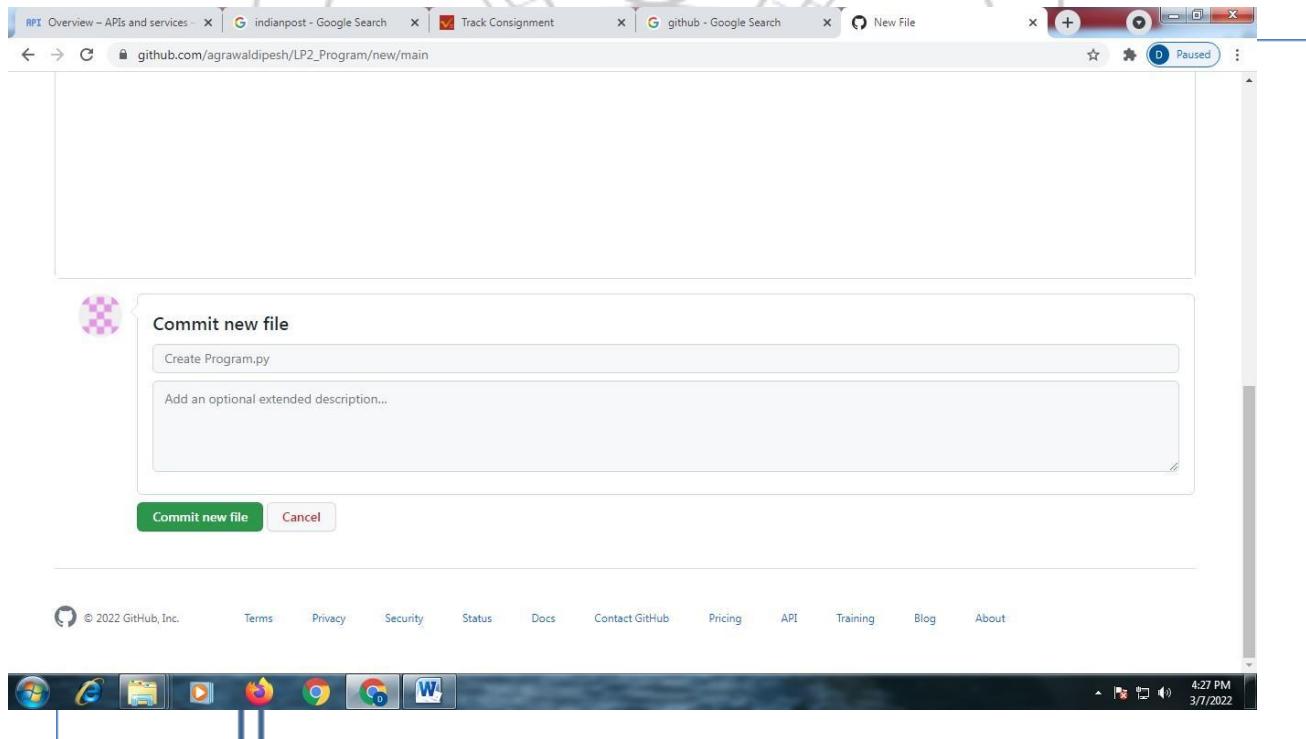
Give any name to the python file, like || Program.py



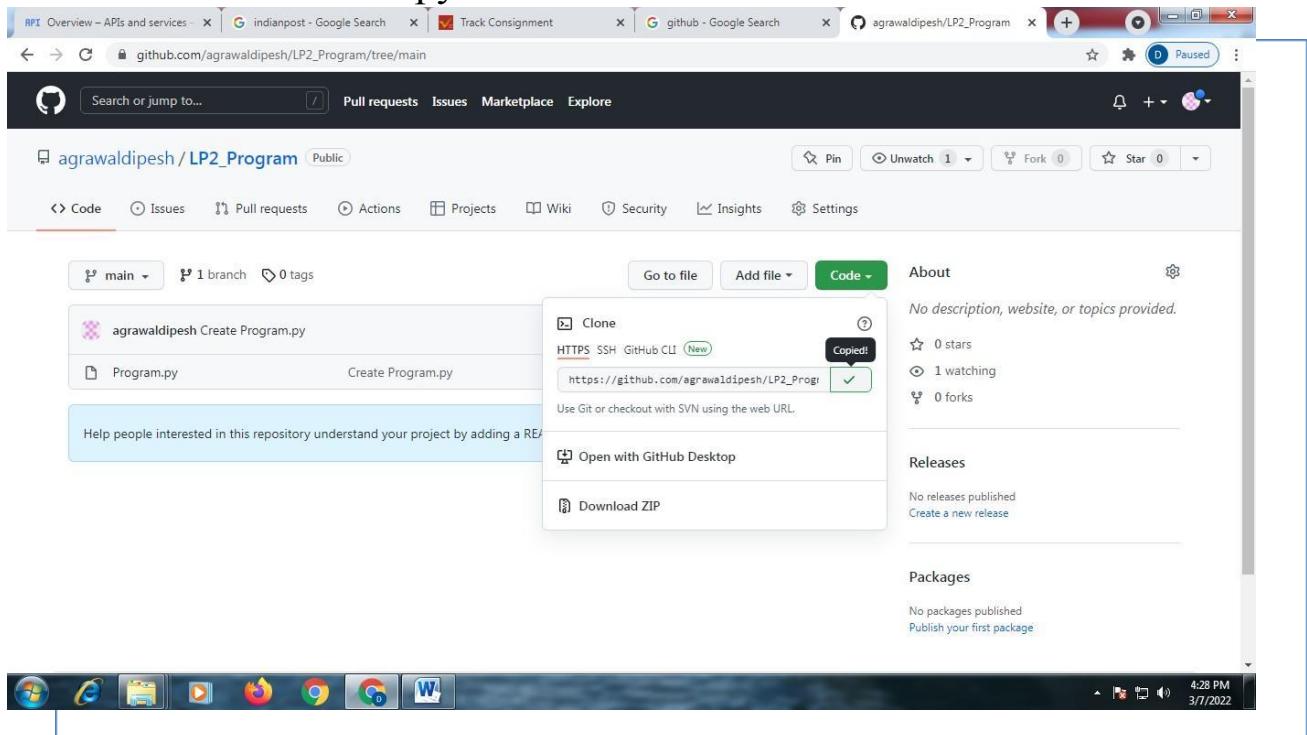
Type your code ↴ Print(—hello||)



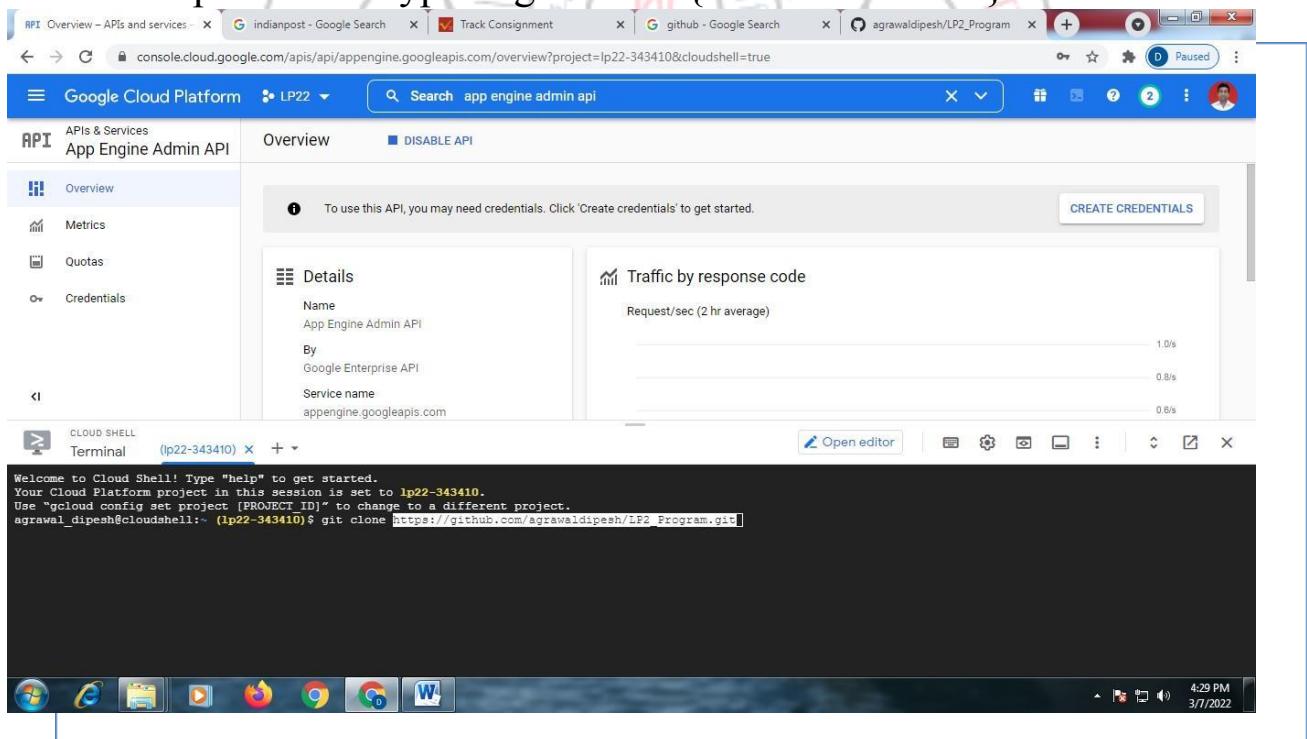
Scroll Down and Click on —Commit new file||



Click on —Code|| and Copy URL □



Go to cloud platform □ type —git clone|| {Paste URL Here} □



Type —ls||

To use this API, you may need credentials. Click 'Create credentials' to get started.

Details

- Name: App Engine Admin API
- By: Google Enterprise API
- Service name: appengine.googleapis.com

Traffic by response code

Request/sec (2 hr average)
1.0/s
0.8/s
0.6/s

CLOUD SHELL

```
Welcome to Cloud Shell! Type "help" to get started.
Your Cloud Platform project in this session is set to lp22-343410.
Use "gcloud config set project [PROJECT ID]" to change to a different project.
agrwal_dipesh@cloudshell:~ (lp22-343410)$ git clone https://github.com/agrawaldipesh/LP2_Program.git
Cloning into 'LP2_Program'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.
agrwal_dipesh@cloudshell:~ (lp22-343410)$ ls
LP2_Program README-cloudshell.txt
agrwal_dipesh@cloudshell:~ (lp22-343410)$
```

Enter into repository, using command — cd repository name (here it is – LP2_Program) — ls

To use this API, you may need credentials. Click 'Create credentials' to get started.

Details

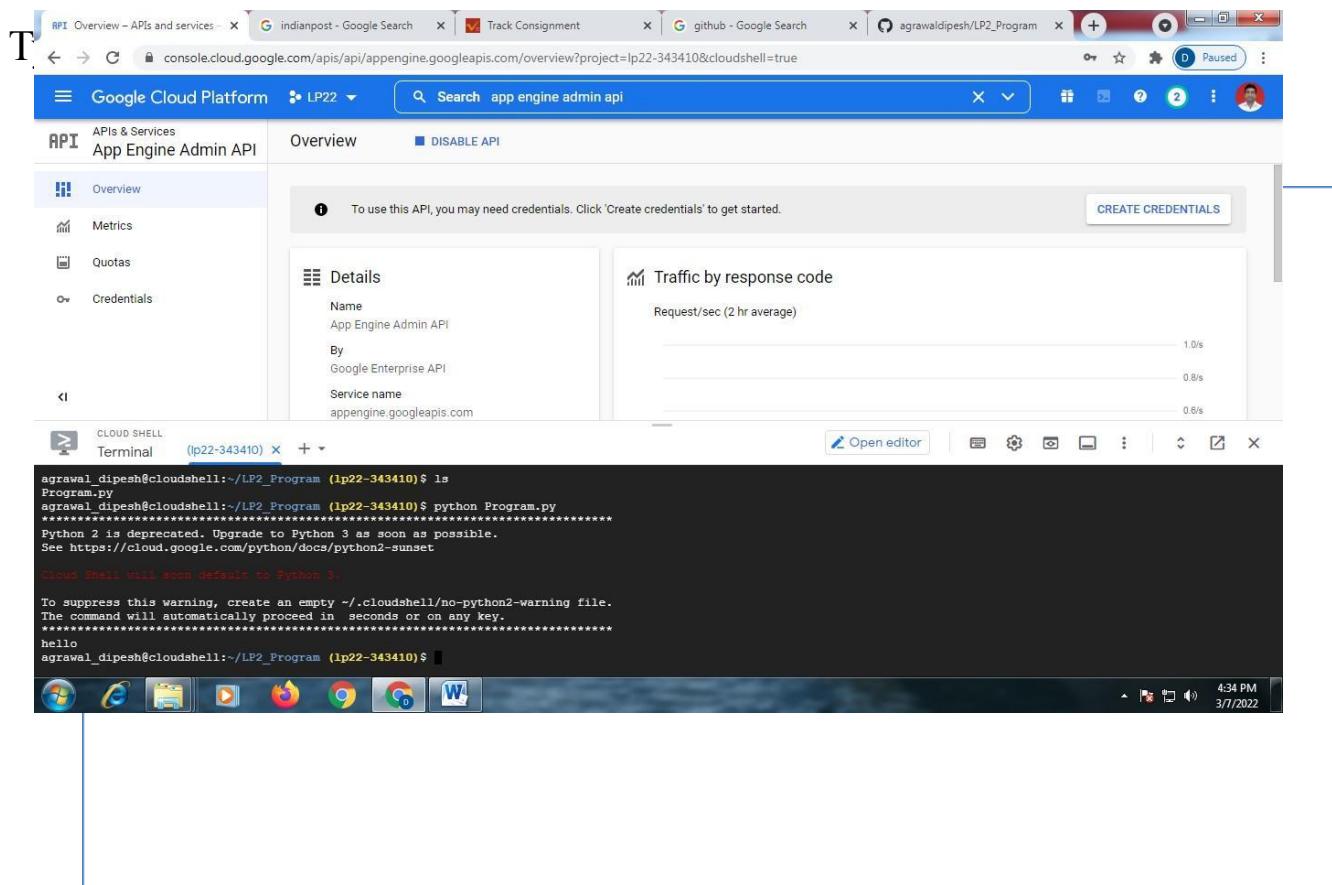
- Name: App Engine Admin API
- By: Google Enterprise API
- Service name: appengine.googleapis.com

Traffic by response code

Request/sec (2 hr average)
1.0/s
0.8/s
0.6/s

CLOUD SHELL

```
Your Cloud Platform project in this session is set to lp22-343410.
Use "gcloud config set project [PROJECT ID]" to change to a different project.
agrwal_dipesh@cloudshell:~ (lp22-343410)$ git clone https://github.com/agrawaldipesh/LP2_Program.git
Cloning into 'LP2_Program'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.
agrwal_dipesh@cloudshell:~ (lp22-343410)$ ls
LP2_Program README-cloudshell.txt
agrwal_dipesh@cloudshell:~ (lp22-343410)$ cd LP2_Program
agrwal_dipesh@cloudshell:/LP2_Program (lp22-343410)$ ls
Program.py
agrwal_dipesh@cloudshell:/LP2_Program (lp22-343410)$
```



6.Frequently Asked Questions:

1. What is Google App Engine ?
2. What is difference between SQL and GQL ?
3. What are the advantages of Google App Engine over Amazon EC2 ?
4. What is Kubernetes ?
5. What is AppScale ?

7.Conclusion:

Successfully Installation and configure Google App Engine

SNJB's Late Sau. K. B. Jain College of Engineering, Chandwad

Department of Computer Engineering

Course Name:Laboratory Practice II(310258):Cloud Computing

Class:Third Year (TE) Div A/ Div B

Batch:T1/T2/T3/T4

Name:

Roll No:

Assignment No: 9

Answers (A) – 5M	Coding Efficiency (C) – 5M	Viva (V) – 5M	Timely Completion (T) – 5M	Total(20M)	Sign

Date of Performance:..... **Date of Completion:**.....

1. Title of Assignment:

Creating an Application in SalesForce.com using Apex programming Language.

2. Objective:

1. To learn the salesforce environment.
2. Create a small application using APEX programming language.

3. Outcome: Use tools and techniques in the area of Cloud Computing

4. Software and Hardware Requirement:

Software Requirement: login required for SalesForce.com

Hardware Requirement: Internet Connection, PC with Min. 2GB RAM, Core i5 Processor

5.Relevant Theory :

What is Salesforce?

Salesforce is the world's #1 cloud-based customer relationship management (CRM) platform. It is an integrated CRM platform that provides a single shared view of each customer for all the departments within an organization, such as Marketing, Sales, Commerce, and Service. Our salesforce tutorial is designed to help beginners with the Salesforce and professionals' basic concepts with advanced concepts. In this, we will cover all the essential topics of Salesforce from beginning to Apex development.

What does Salesforce do?

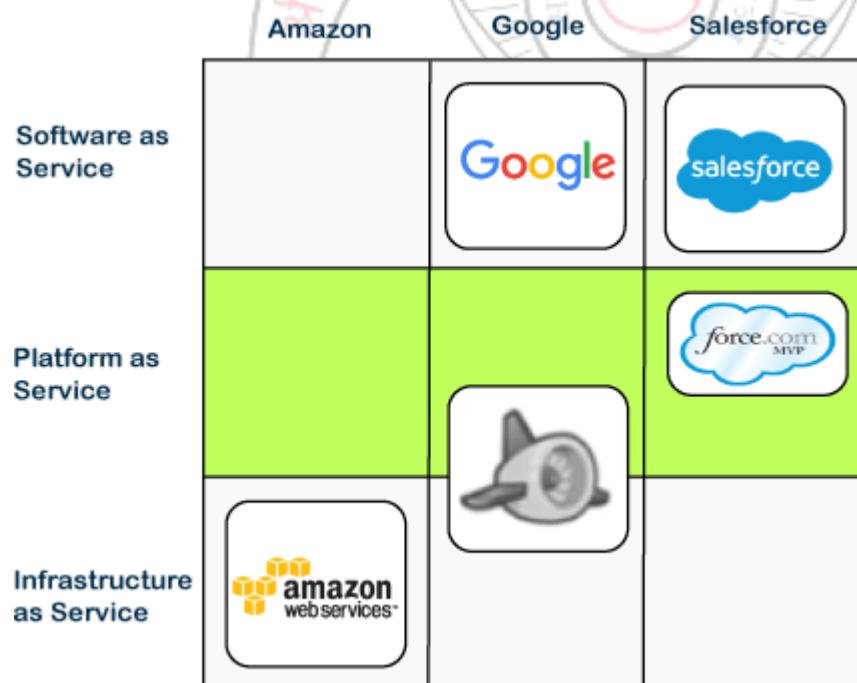
Salesforce bring companies and customers together. Salesforce unites your marketing, sales, commerce, service, and IT teams from anywhere with Customer 360 — one integrated CRM platform that powers our entire suite of connected apps. With Customer 360, you can focus your employees on what's important right now: stabilizing your business, reopening, and getting back to delivering exceptional customer experiences.

What is Salesforce used for?

1. Attract more buyers using personalized marketing.
2. Win more customers by getting to know their needs and concerns.
3. Deliver the amazing shopping experiences your customers expect.
4. Respond faster to customer support issues on any channel.
5. Automate time-consuming tasks by building custom apps.

Over 150,000 companies, both big and small, are growing their business with Salesforce.

- o Salesforce is a SaaS or Software as a Service, which means there is no need to install the software or server to work on. Users can simply sign-up in Salesforce.com and can start running the business instantly.



- o It was founded by Marc Benioff, Parker Harris, Dave Moellenhoff, and Frank Dominguez in 1999.
- o Salesforce was started as a CRM software, but today it provides various products and software solutions to users and developers.
- o Since Salesforce is cloud-based software, hence it does not require any IT

professional to set up anything. It provides one of the best ways to connect with customers, business partners, and clients over the single integrated environment. It allows the businesses to identify the customer's requirements, address the problems easily, and provide the same solution in the minimum timeframe.

Companies using Salesforce

Today, there are multiple big brands, and new start-ups who are using the Salesforce platform as CRM software and for other services. Below is the list of some popular companies that are using Salesforce for their businesses:

1. HCL Technologies use Salesforce CRM for data entry to validate customer data.
2. Pizza Hut is providing the smart mobile experience to its customer using the marking salesforce cloud.
3. L'Oreal builds a social network using the Salesforce, by which they can share any Hair fashion-related plans or events to their stylist or customer.
4. American Express is using the Salesforce cloud since 2010, and now they can connect to their employees together across the organization, or from anywhere.
5. Nikon Instrument opted for the Salesforce CRM for their business, but later, they also started using the social network provided by the Chatter.
6. Comcast-Spectator uses the salesforce platform to manage the customer profiles that help them identify the customer interest and market more efficiently.
7. Sony uses the Salesforce Service Cloud to connect with their customers. All customer complaints are managed and taken care of with this to make the customers satisfied.

Other companies, such as Urban Ladder, InMobi, Paytm, Genesys, etc., use Salesforce services for their business growth.

What is Apex?

Apex is a proprietary language developed by the Salesforce.com. As per the official definition, Apex is a strongly typed, object-oriented programming language that allows developers to execute the flow and transaction control statements on the Force.com platform server in conjunction with calls to the Force.com API.

It has a Java-like syntax and acts like database stored procedures. It enables the 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. Apex is included in Performance Edition, Unlimited Edition, Enterprise Edition, and Developer Edition.

When Should Developer Choose Apex?

Apex should be used when we are not able to implement the complex business functionality using the pre-built and existing out of the box functionalities. Below are the cases where we need to use apex over Salesforce configuration.

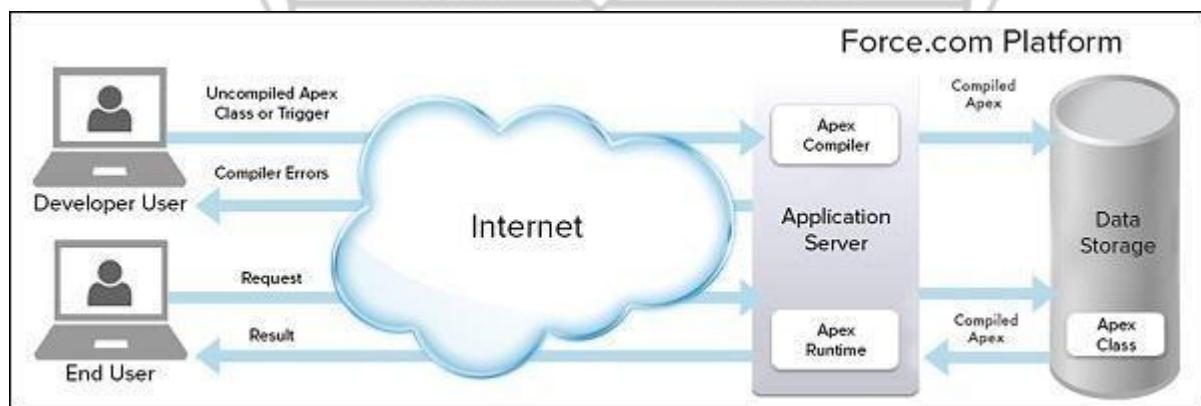
Apex Applications

We can use Apex when we want to –

- Create Web services with integrating other systems.
- Create email services for email blast or email setup.
- Perform complex validation over multiple objects at the same time and also custom validation implementation.
- Create complex business processes that are not supported by existing workflow functionality or flows.
- Create custom transactional logic (logic that occurs over the entire transaction, not just with a single record or object) like using the Database methods for updating the records.
- Perform some logic when a record is modified or modify the related object's record when there is some event which has caused the trigger to fire.

Working Structure of Apex

As shown in the diagram below (Reference: Salesforce Developer Documentation), Apex runs entirely on demand Force.com Platform



Flow of Actions

There are two sequence of actions when the developer saves the code and when an end user performs some action which invokes the Apex code as shown below –

Developer Action

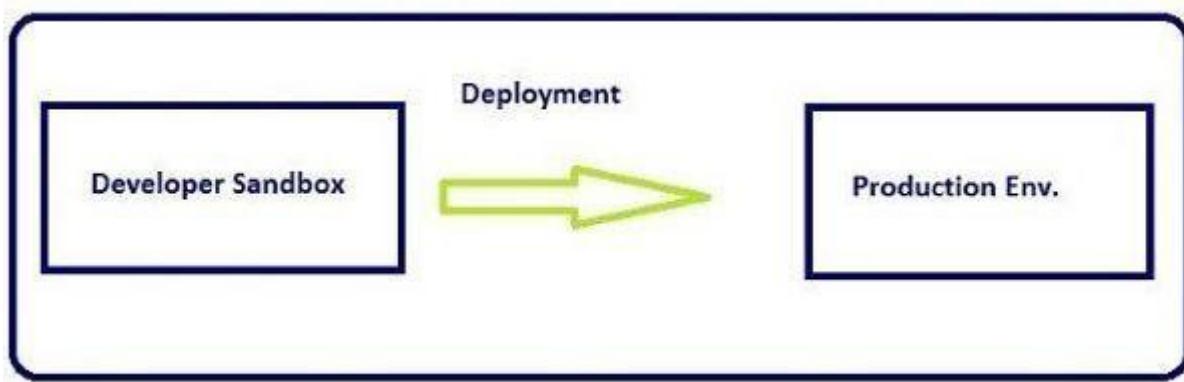
When a developer writes and saves Apex code to the platform, the platform application server first compiles the code into a set of instructions that can be understood by the Apex runtime interpreter, and then saves those instructions as metadata.

End User Action

When an end-user triggers the execution of Apex, by clicking a button or accessing a Visualforce page, the platform application server retrieves the compiled instructions from the metadata and sends them through the runtime interpreter before returning the result. The end-user observes no differences in execution time as compared to the standard application platform request.

Since Apex is the proprietary language of Salesforce.com, it does not support some features which a general programming language does. Following are a few features which Apex does not support –

- It cannot show the elements in User Interface.
- You cannot change the standard SFDC provided functionality and also it is not possible to prevent the standard functionality execution.
- Creating multiple threads is also not possible as we can do it in other languages.
- You can develop the Apex code in either Sandbox or Developer edition of Salesforce. A Sandbox organization is a copy of your organization in which you can write code and test it without taking the risk of data modification or disturbing the normal functionality. As per the standard industrial practice, you have to develop the code in Sandbox and then deploy it to the Production environment.
- For this tutorial, we will be using the Developer edition of Salesforce. In the Developer edition, you will not have the option of creating a Sandbox organization. The Sandbox features are available in other editions of Salesforce.



Apex Code Development Tools

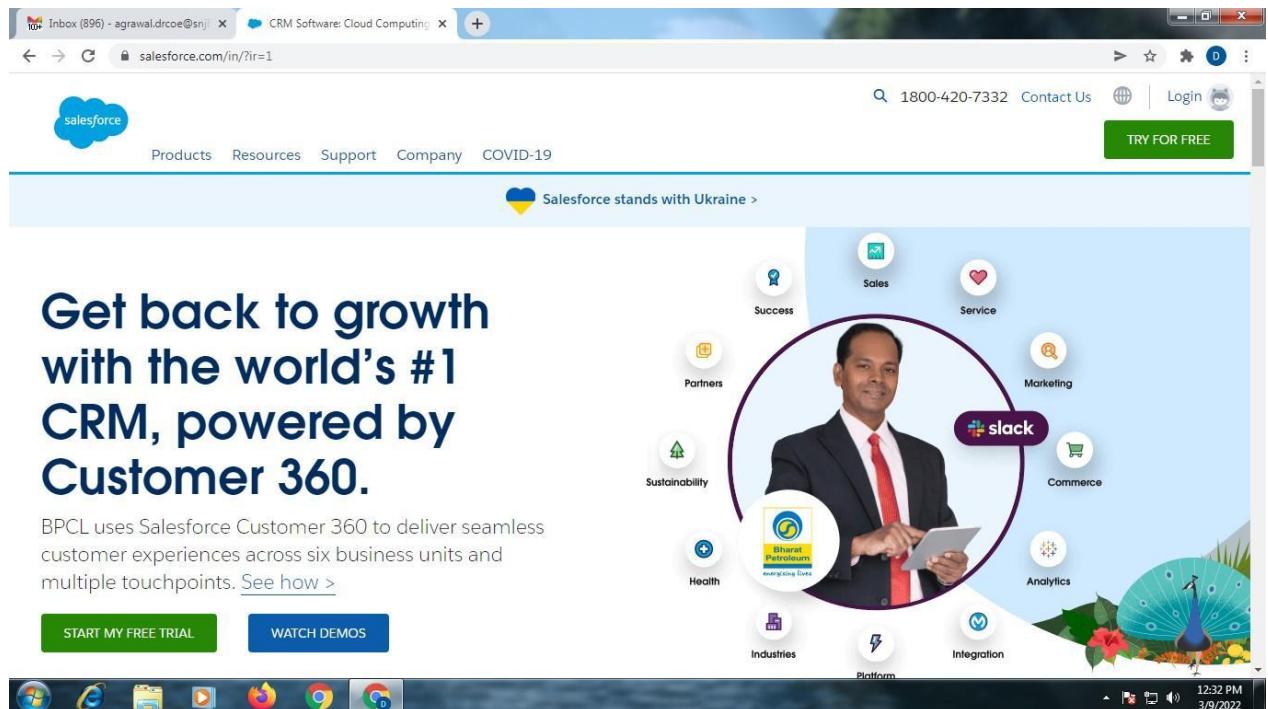
In all the editions, we can use any of the following three tools to develop the code –

- Force.com Developer Console
- Force.com IDE
- Code Editor in the Salesforce User Interface

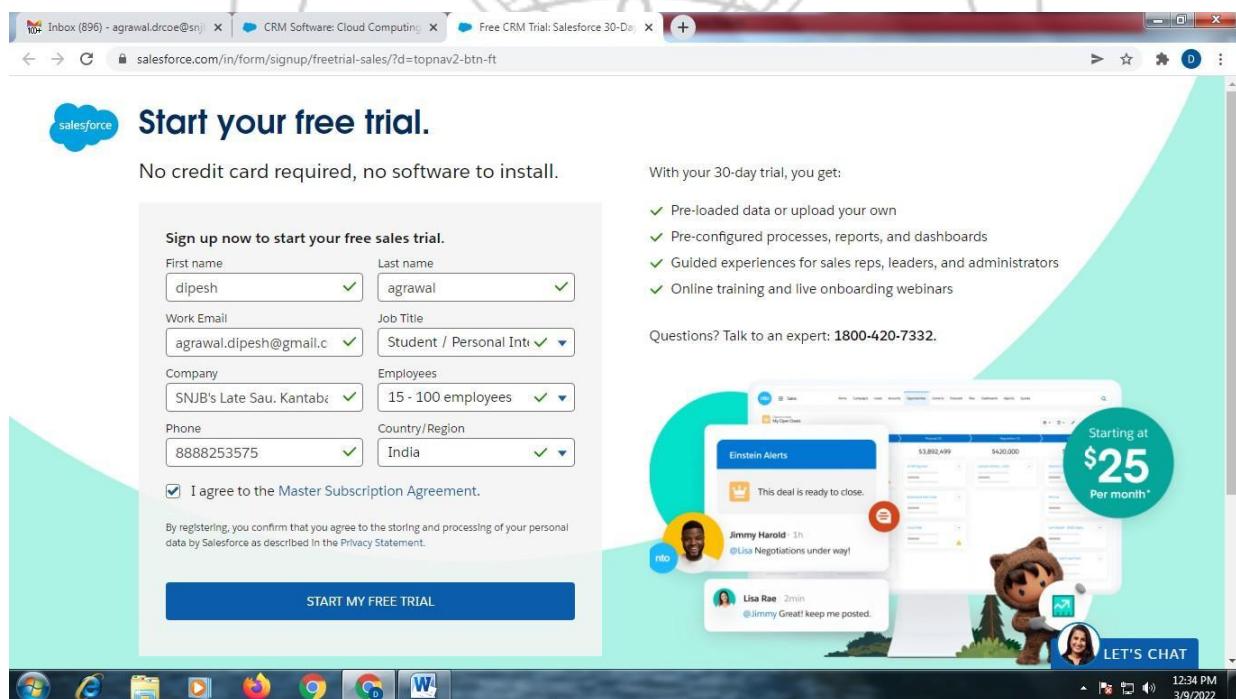
Note – We will be utilizing the Developer Console throughout our tutorial for code execution as it is simple and user friendly for learning.

Steps –

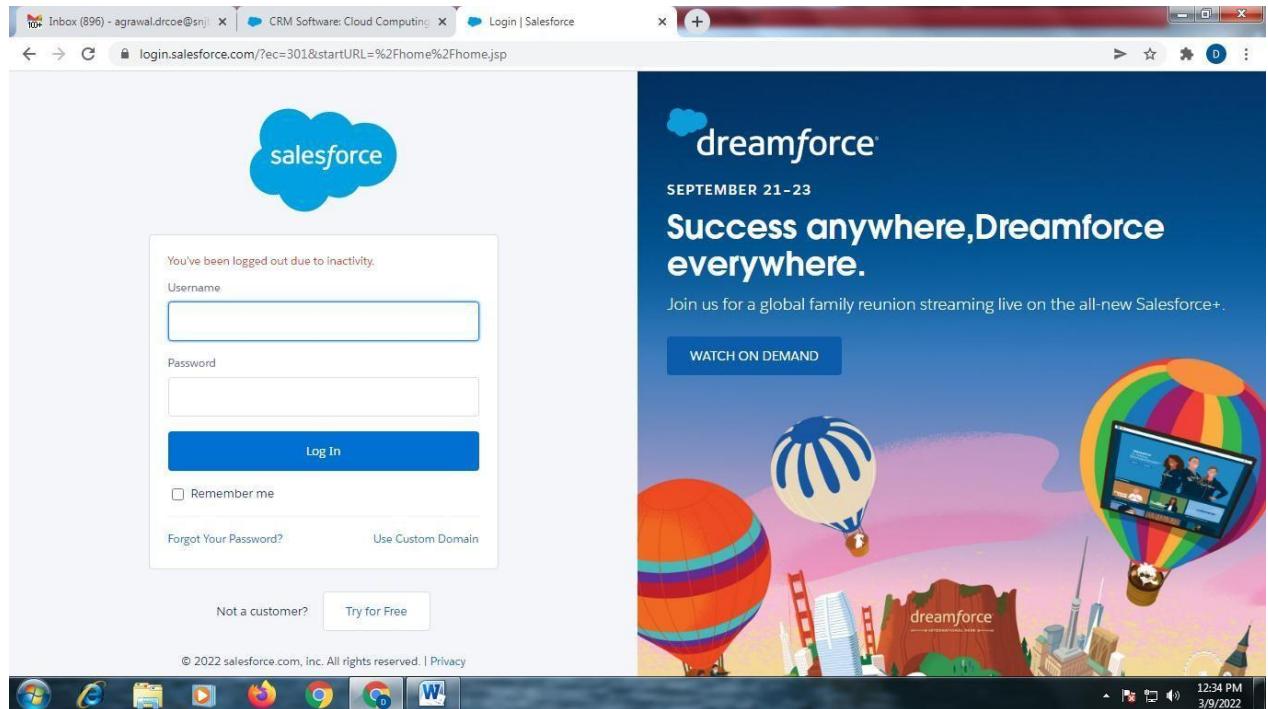
Open www.salesforce.com website in browser.



Create Your Account ?



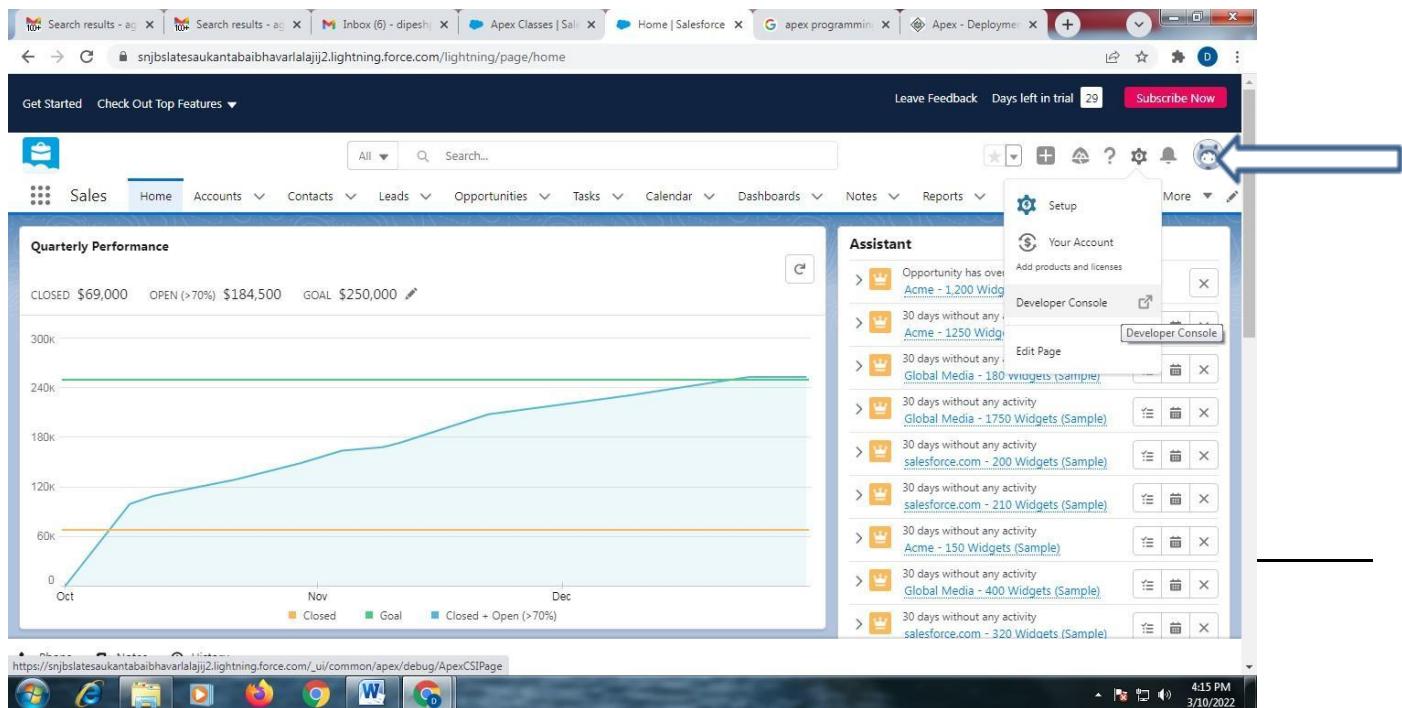
Enter Login-id and Password



Following Screen will appear



Click on “Setup”

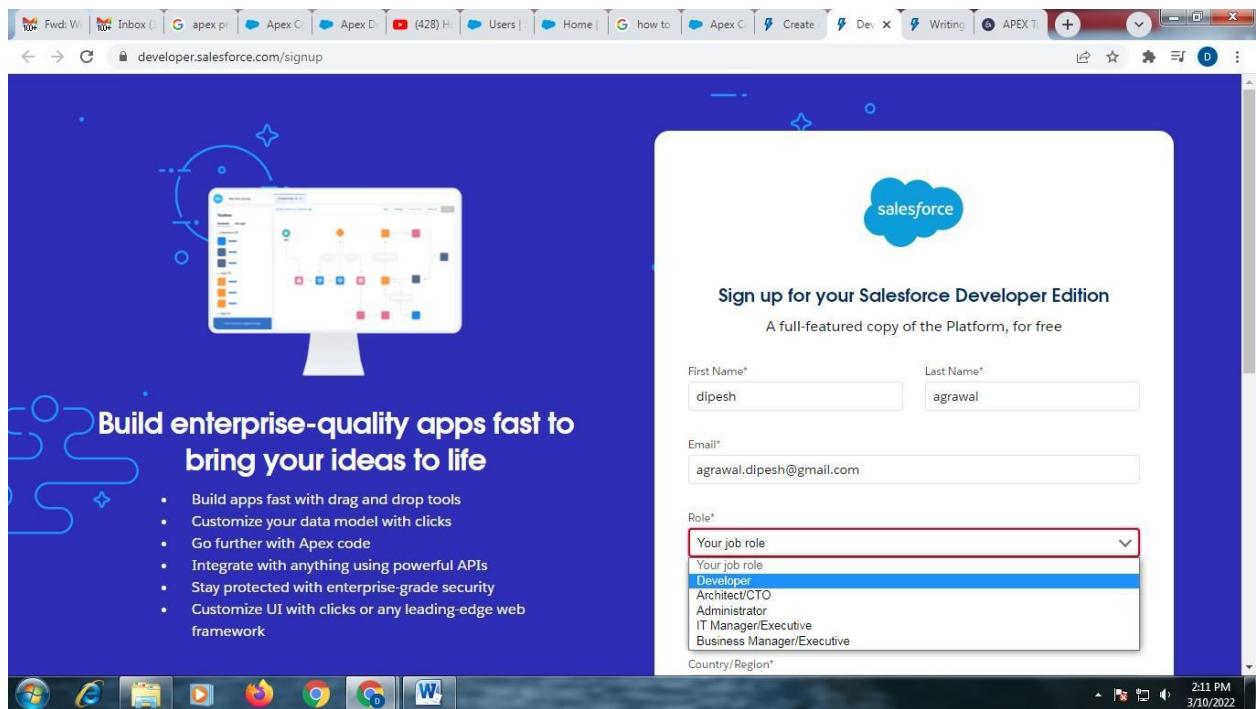


Click on “Developer Console”

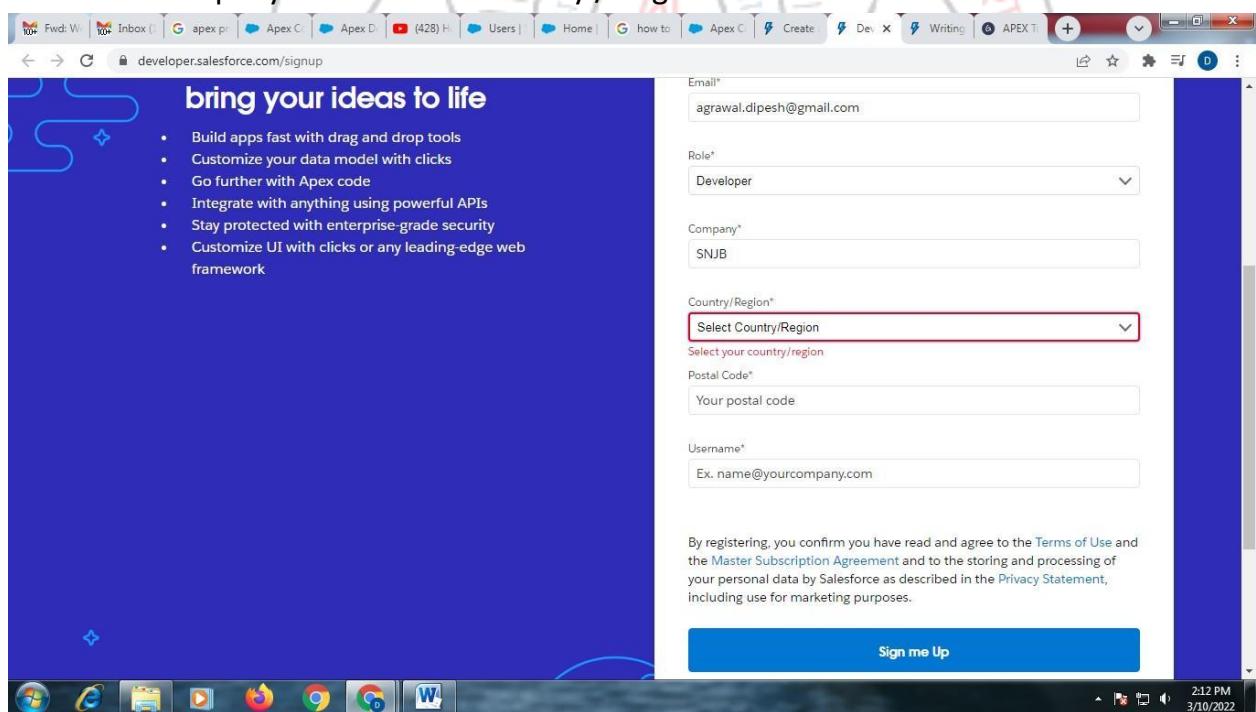
Create Account on Developer Console, using following steps Developer.salesforce.com/signup



Enter your First Name, Last Name and Email Select your Job Role as – Developer



Enter Your Company Name Select Country / Region



Enter Your Postal Code

Enter user name ↗ different than your Email-id

For ex. I have used email id for my salesforce account as ↗

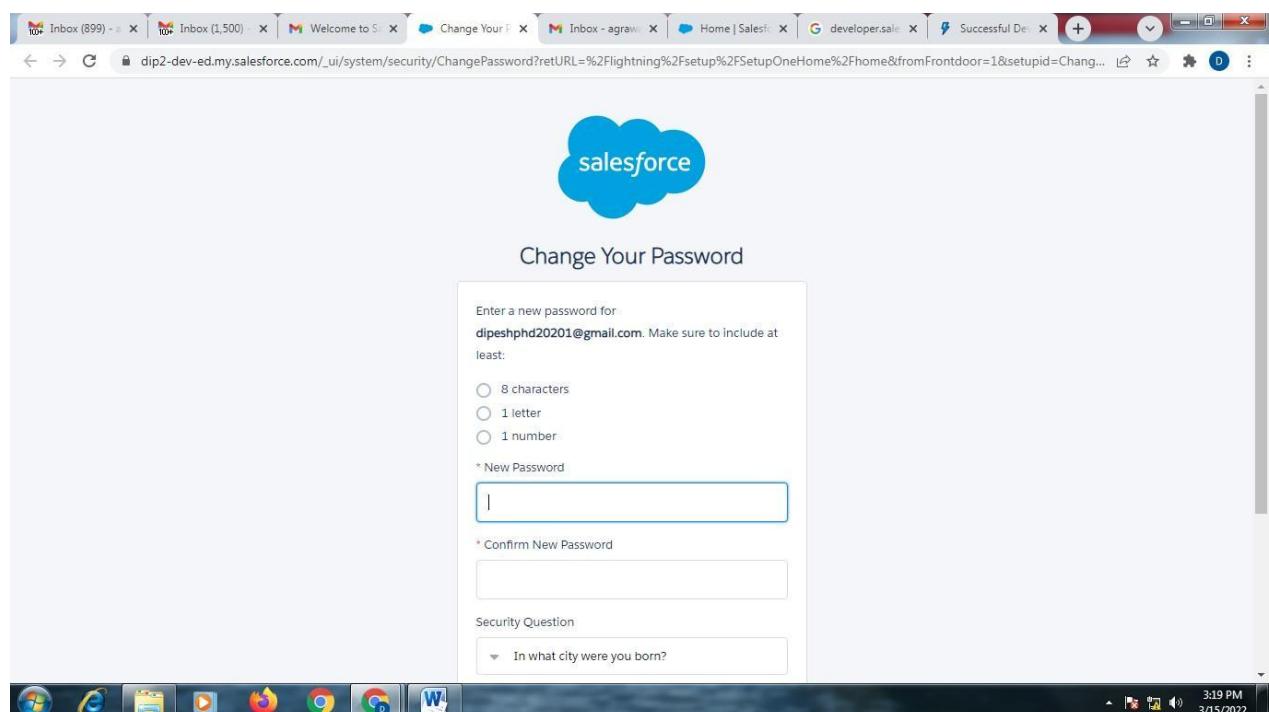
dfgdg.ghjghj@gmail.com,

Then, for Developer account, I will use different email-id as

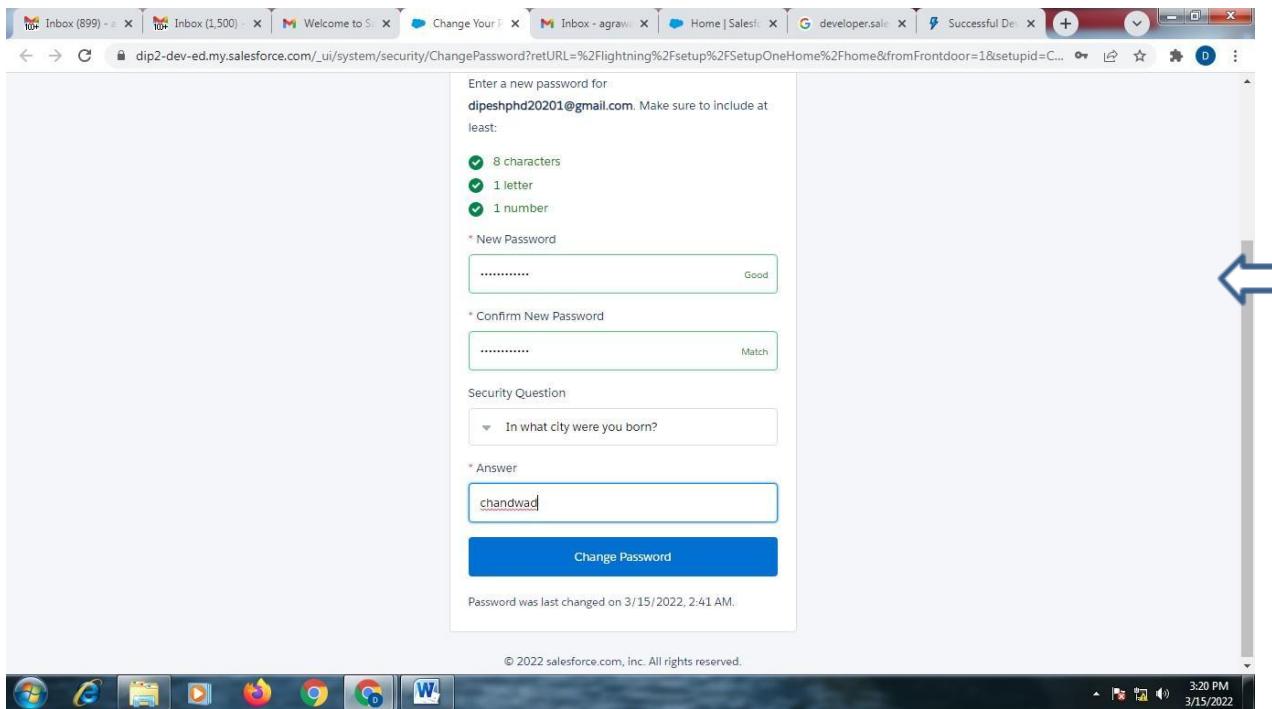
↗ jghjghj.fujifuiuy@gmail.com

You will get an email of verification link in your email account, as below ↗ click on “Verify your account”

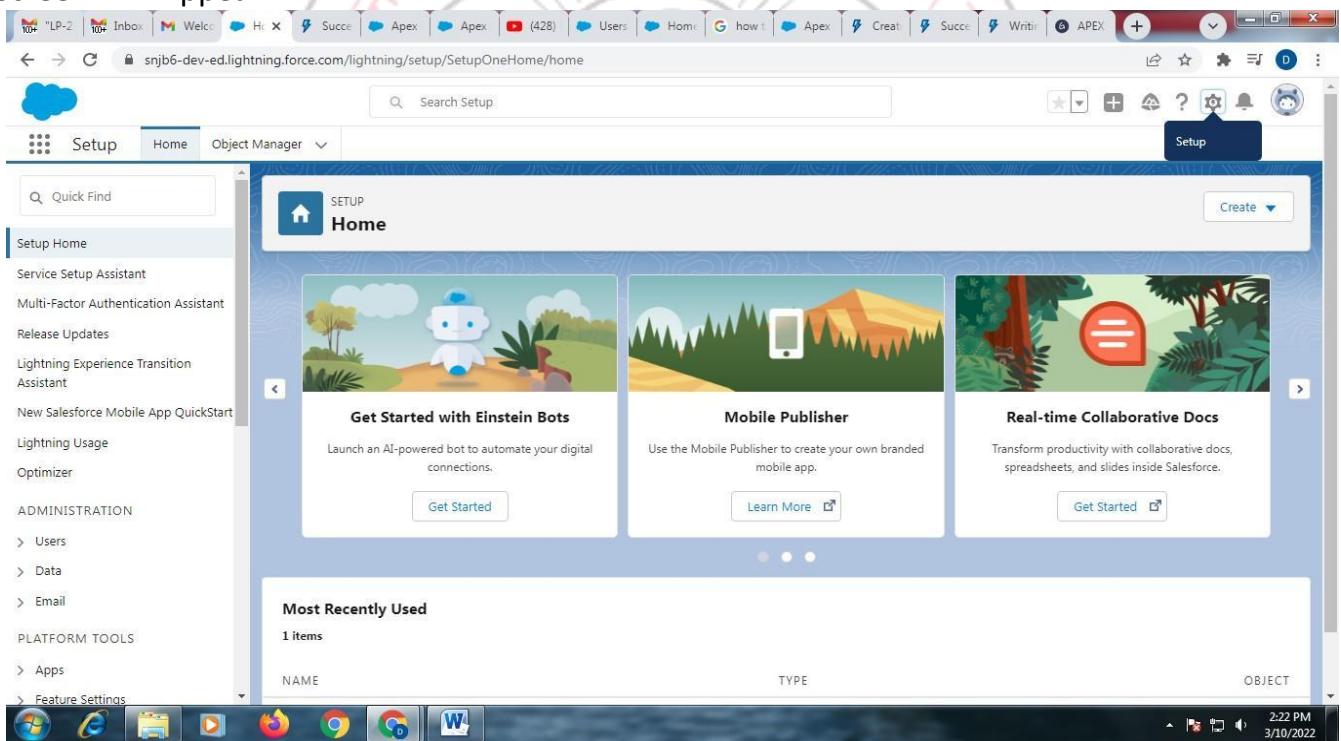
Following Screen Will Appear ↗



ESTD - 1928



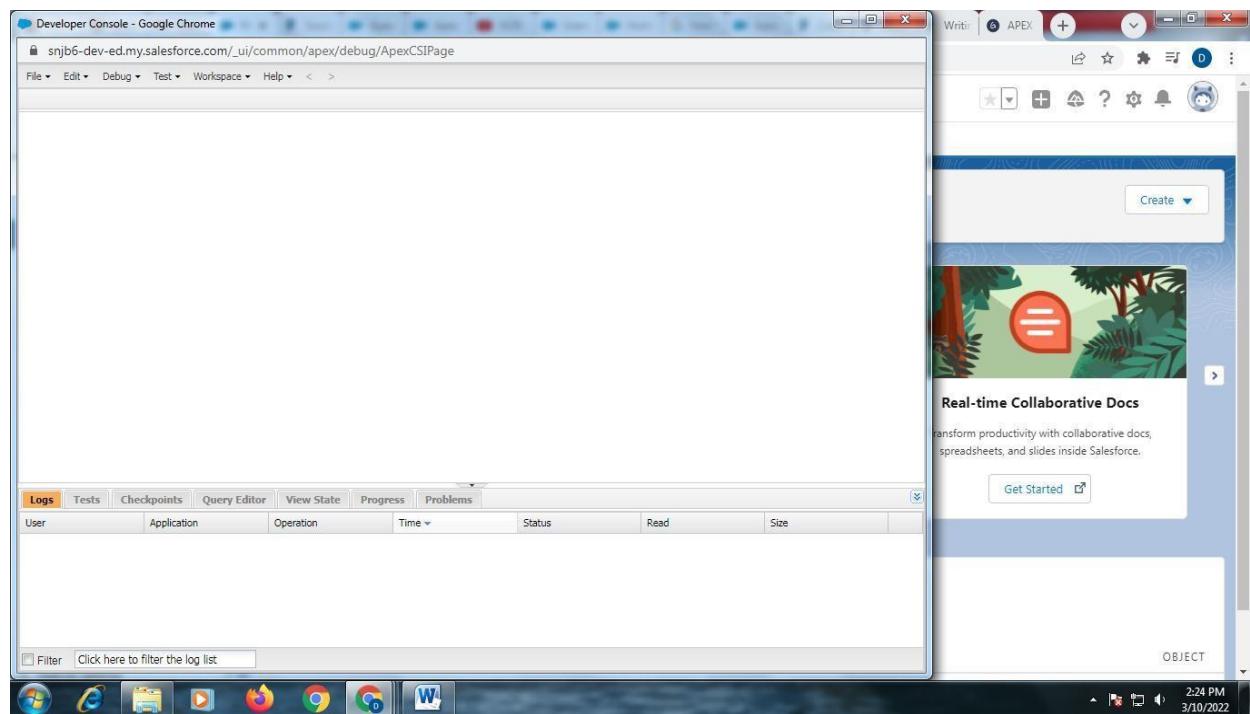
Enter Password, Security Question and Answer of itClick on “Change Password” Following Screen Will Appear ↴



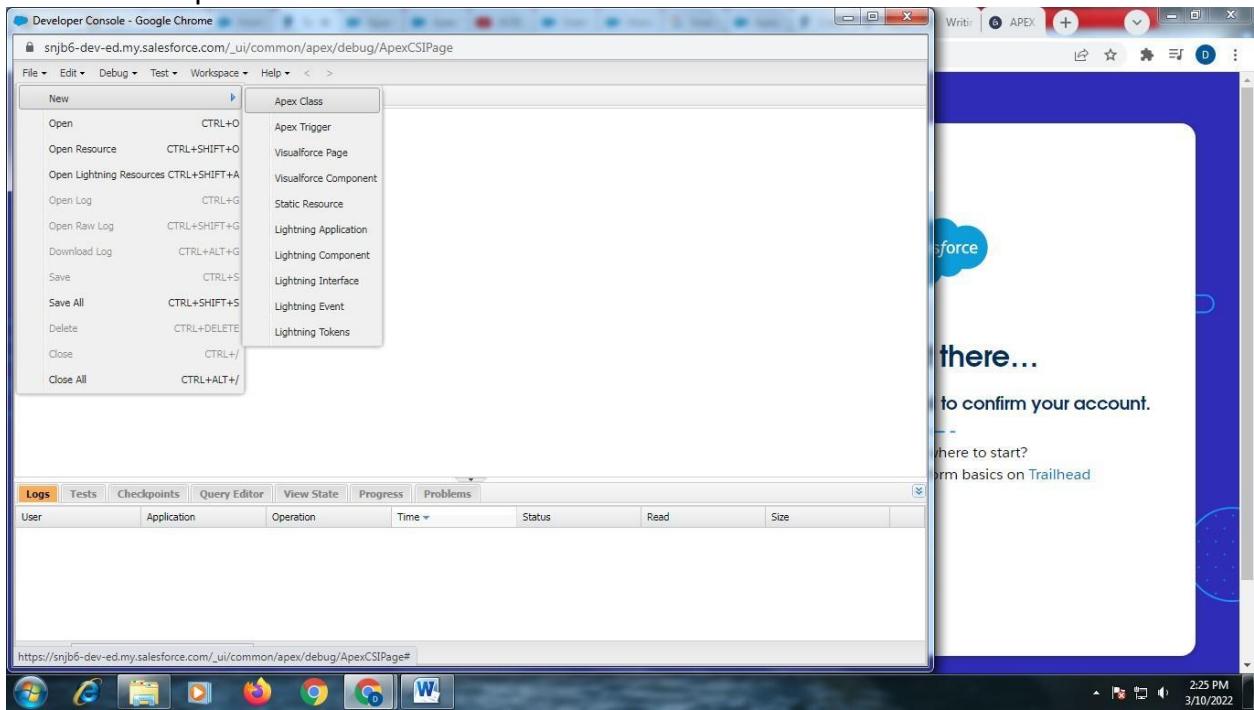
Click on Setup

Select Developer Console

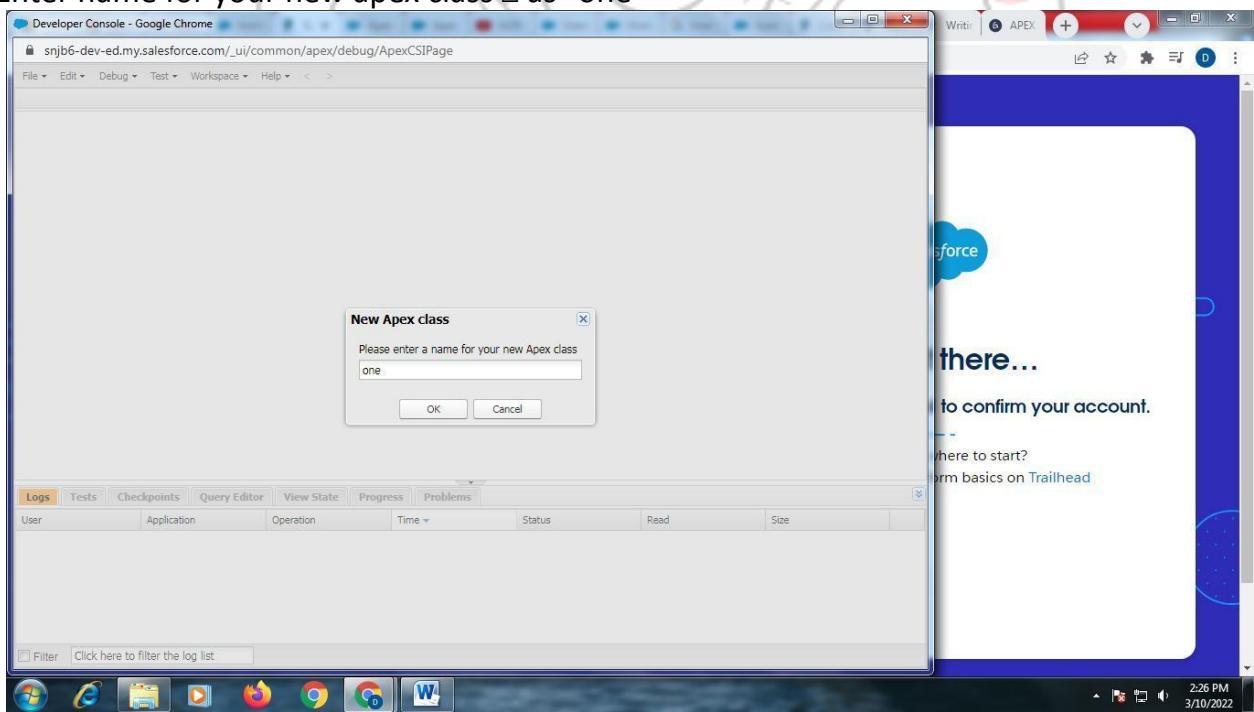
Developer Console will be opened as below



File ▾ New ▾ Apex Class

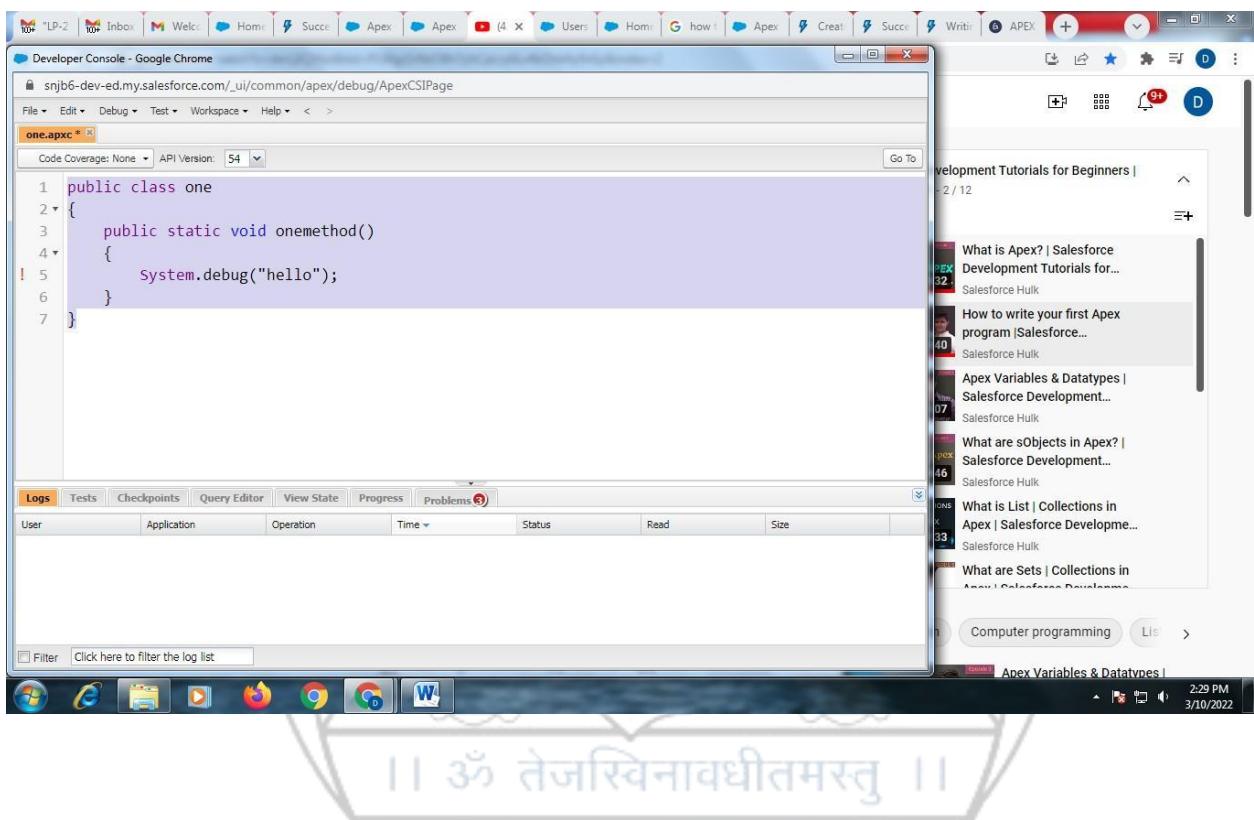


Enter name for your new apex class ▾ as "one"



Write Following Code ↴

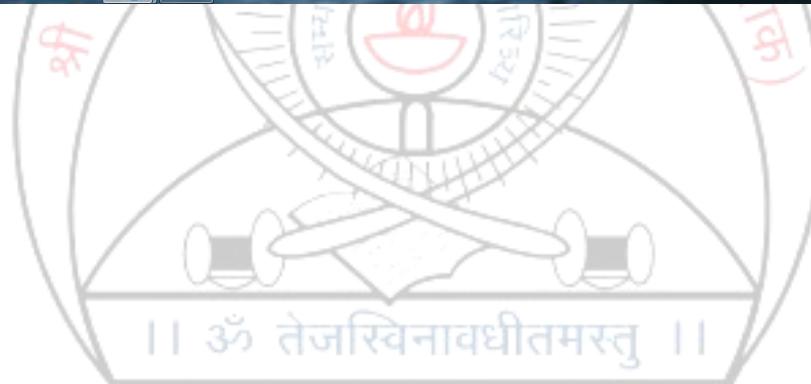
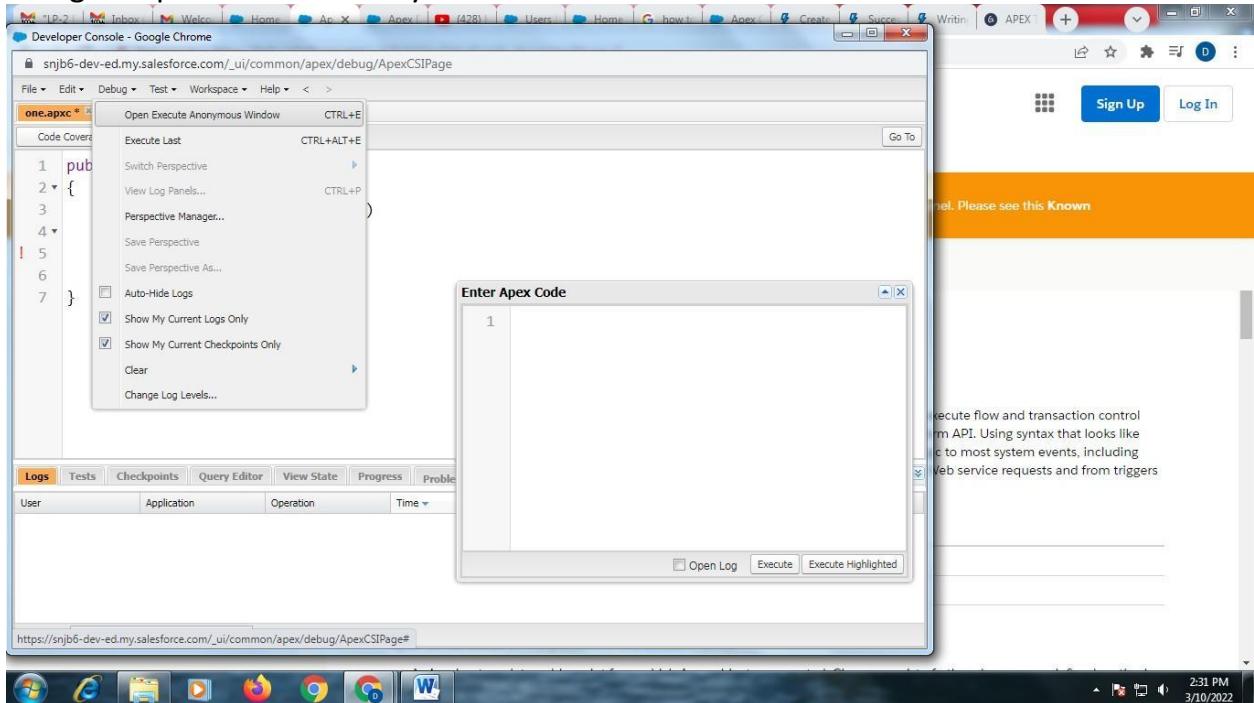
```
public class one
{
    public static void onemethod()
    {
        System.debug("hello");
    }
}
```



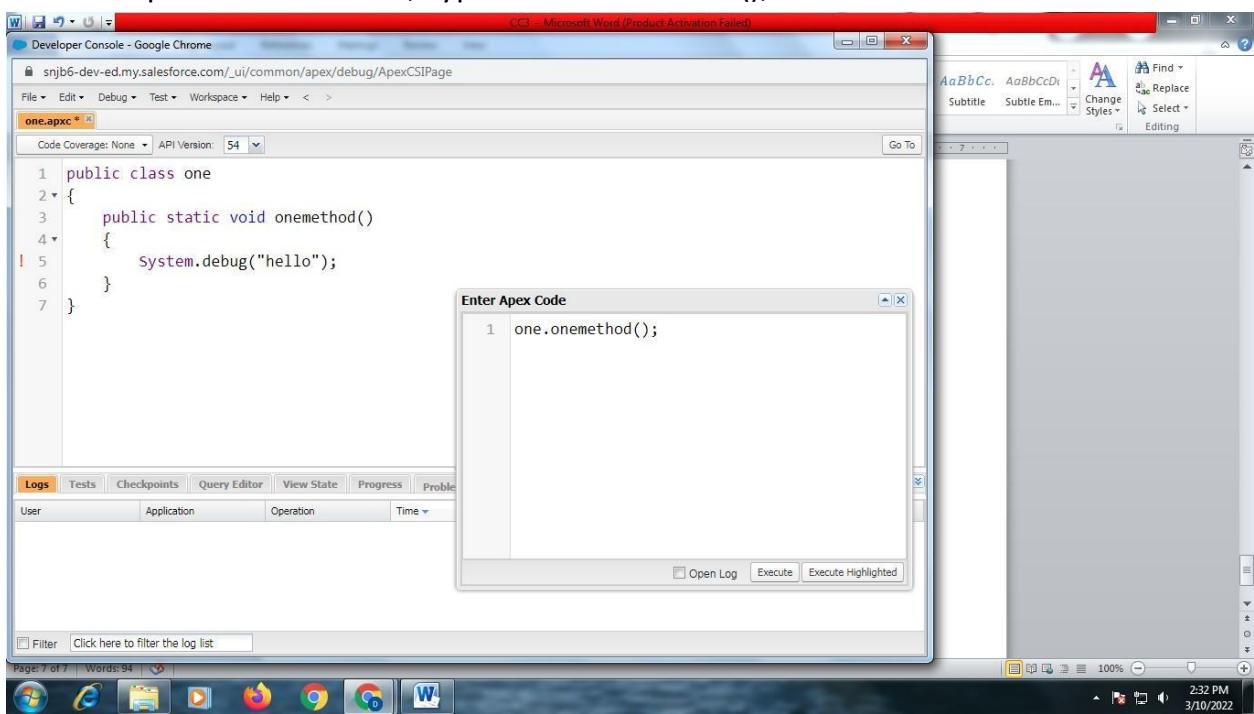
Save the code using File → Save

Debug the code using following steps ↗

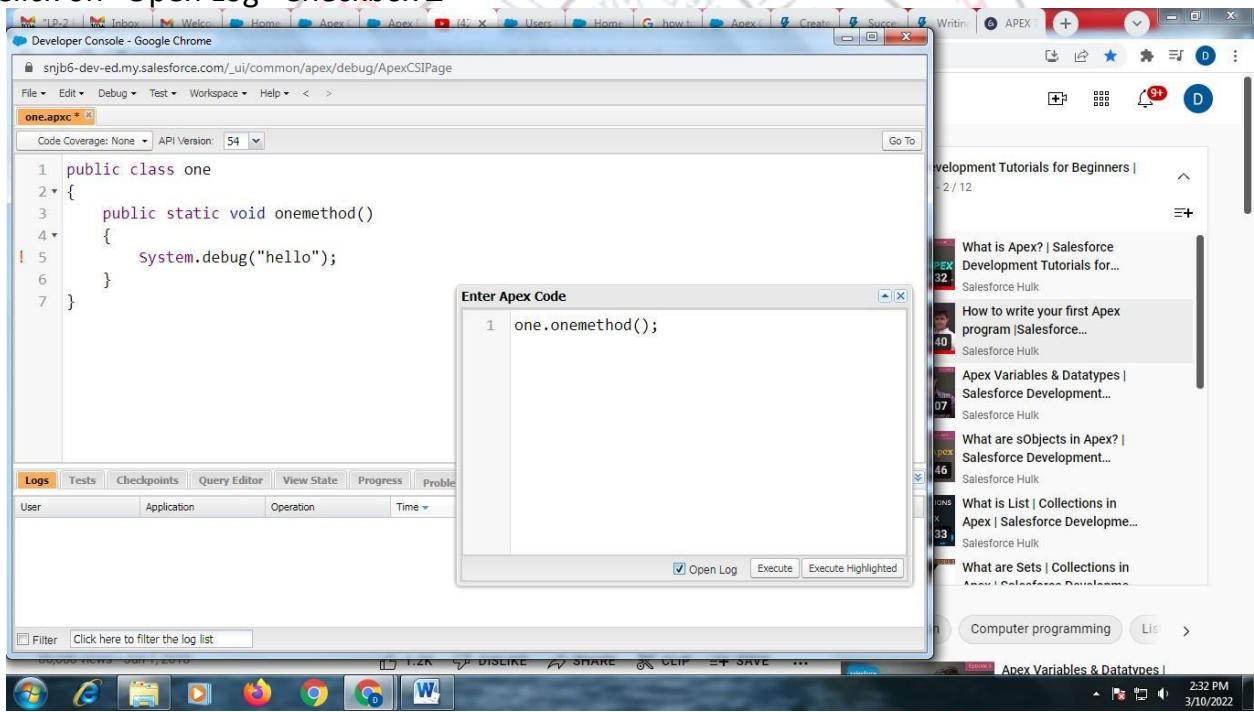
Debug ↗ “Open Execute Anonymous Window”



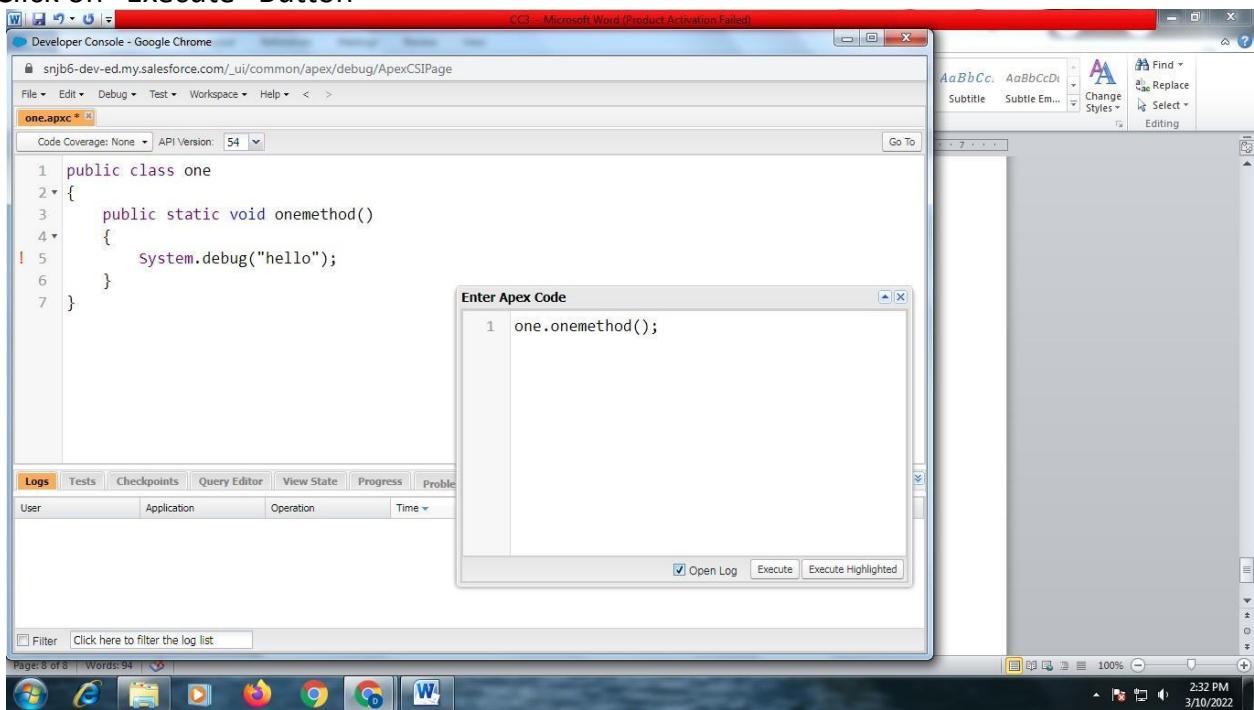
In “Enter Apex Code” window, Type “one.onemethod();”



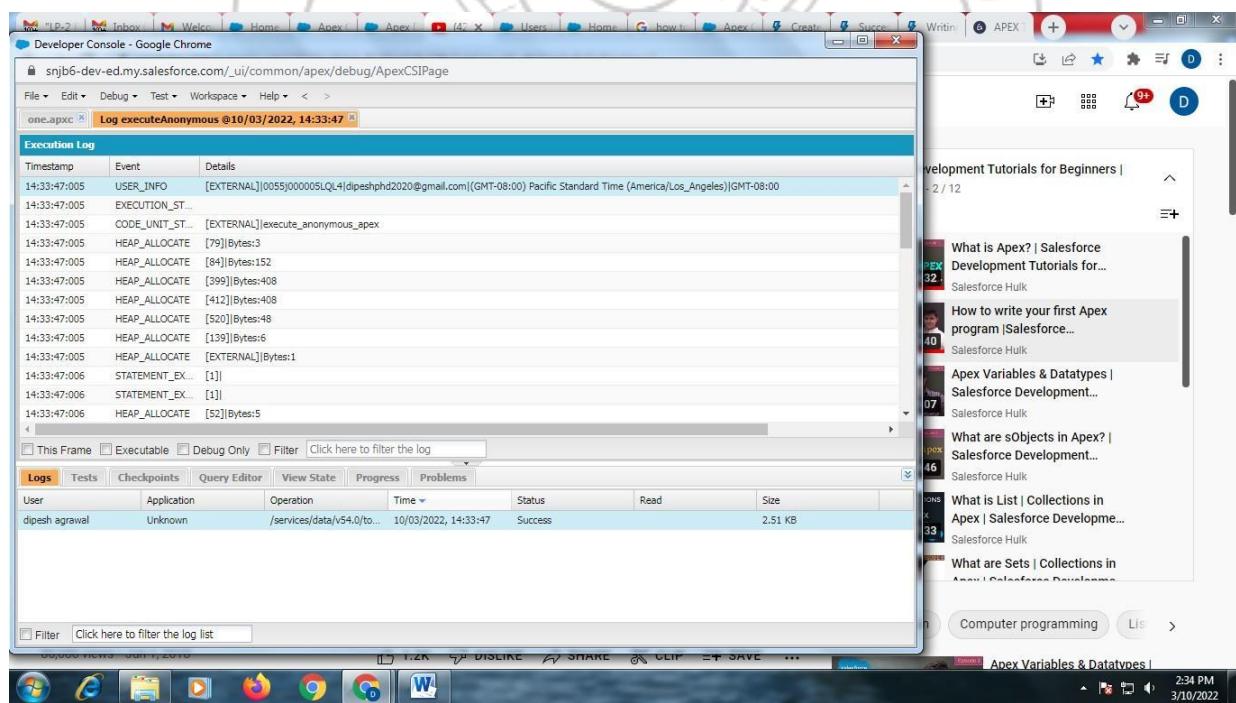
Click on “Open Log” Checkbox



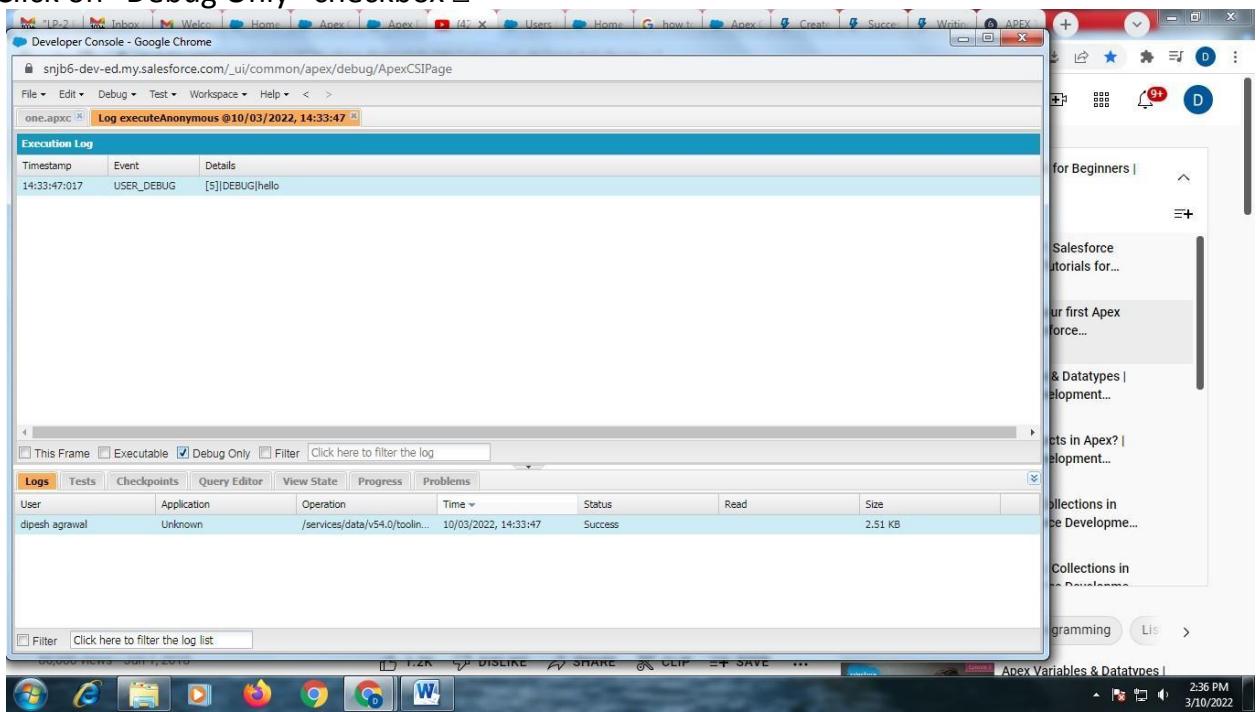
Click on “Execute” Button



Debug Log will be opened in the following window



Click on “Debug Only” checkbox ☰



Observe the output in ☰ “Execution Log”

Assignment Questions ☰

1. What is Salesforce?
2. What does Salesforce do?
3. What is Salesforce used for?
4. What is Apex?
5. When Should Developer Choose Apex?

Conclusion ☰

Created an Application in SalesForce.com using Apex programming Language.

SNJB's Late Sau. K. B. Jain College of Engineering, Chandwad

Department of Computer Engineering

Course Name:Laboratory Practice II(310258):Cloud Computing

Class:Third Year (TE) Div A/ Div B

Batch:T1/T2/T3/T4

Name:

Roll No:

Assignment No: 10

Answers (A) – 5M	Coding Efficiency (C) – 5M	Viva (V) – 5M	Timely Completion (T) – 5M	Total(20M)	Sign

Date of Performance:..... **Date of Completion:**.....

1. Title of Assignment:

Design and develop custom Application (Mini Project) using Sales force Cloud.

2. Objective:

1. To Learn & develop the application in Salesforce.com
2. Create Application in Salesforce

ESTD - 1928

3. Outcome: Design and develop applications on cloud

4. Software and Hardware Requirement:

Software Requirement: login required for SalesForce.com

Hardware Requirement: Internet Connection, PC with Min. 2GB RAM, Core i5 Processor

5.Relevant Theory :

What is Salesforce?

Salesforce is the world's best cloud-based customer relationship management (CRM) platform. It is an integrated CRM platform that provides a single shared view of each customer for all the departments within an organization, such as Marketing, Sales, Commerce, and Service.

The Cloud Services That Are Offered By Salesforce Are:



Salesforce Sales Cloud – The Sales Cloud is a CRM platform that enables you to manage your organization's sales, marketing, and customer support facets. If your company is engaged in business-to-business (B2B) and business-to-customer (B2C), then a sales cloud is the service your sales team needs.



Salesforce Marketing Cloud – The marketing cloud provides you with one of the world's most powerful digital marketing platforms. The marketers in your organization can use it to manage customer journey, email, mobile, social media, web personalization, content creation, content management, and data analytics.



Salesforce Service Cloud – The Service Cloud is a service platform for your organization's customer service and support team. It provides features like case tracking and social networking plug-in for conversation and analytics. This not only helps your agents to solve customer problems faster but also gives your customers access to answers. Using these answers your customers can solve problems on their own.



Salesforce Community Cloud – If you need a social platform for your organization to connect and facilitate communication among your employees, partners and customers then Salesforce Community Cloud is the service you need. You can use this platform to exchange data and images in real-time.



Salesforce Commerce Cloud – The commerce cloud enables your organization to provide seamless customer service and experience irrespective of your customer's location (online or in-store). It also provides for customer data integration so that your consumers can have a better experience. If your goal is to provide customers with a positive, engaging customer experience, Commerce Cloud is the service you need.



Salesforce Analytics Cloud – The Analytics Cloud provides a business intelligence platform for your organization to work with large data files, create graphs, charts and other pictorial representations of data. It is optimized for mobile access and data visualization and can be integrated with other Salesforce clouds.



Salesforce App Cloud – To develop custom apps that will run on the Salesforce platform, you can use the Salesforce App Cloud. It provides you with a collection of development tools that you can utilize to create custom applications. Some of the tools in the App Cloud include:

ESTD - 1928

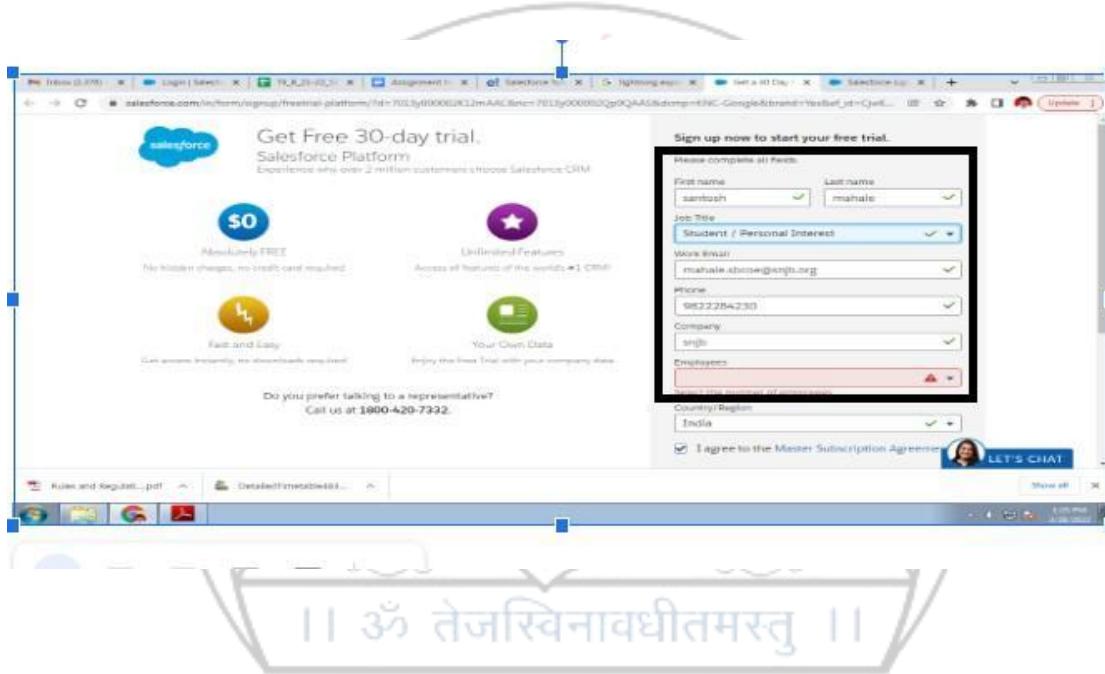
What Is Custom Application Development?

Custom software application development is the process by which a company is able to design personalized software solutions for a specific user or group of users within their organization. The resultant applications will be able to address company needs more effectively and precisely than off-the-shelf options. However, the time, money, and talent needed to design and implement these custom applications have historically been prohibitive. Only the most successful organizations could afford to create their own app solutions.

Steps in Details:

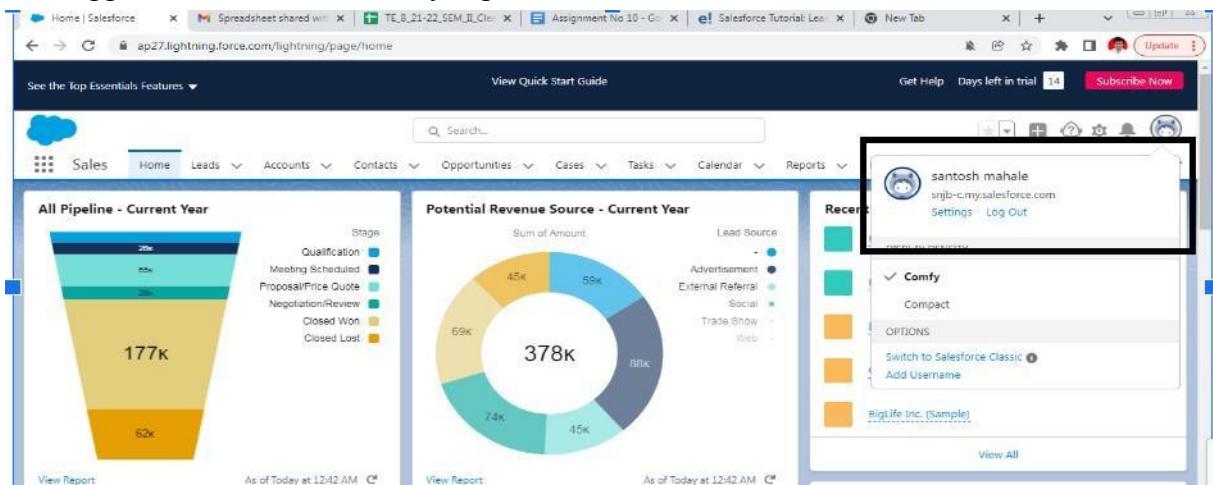
Steps for creating & Starting Saleceforce.com applcaiton

Step-1: Account Creation in Salesforce

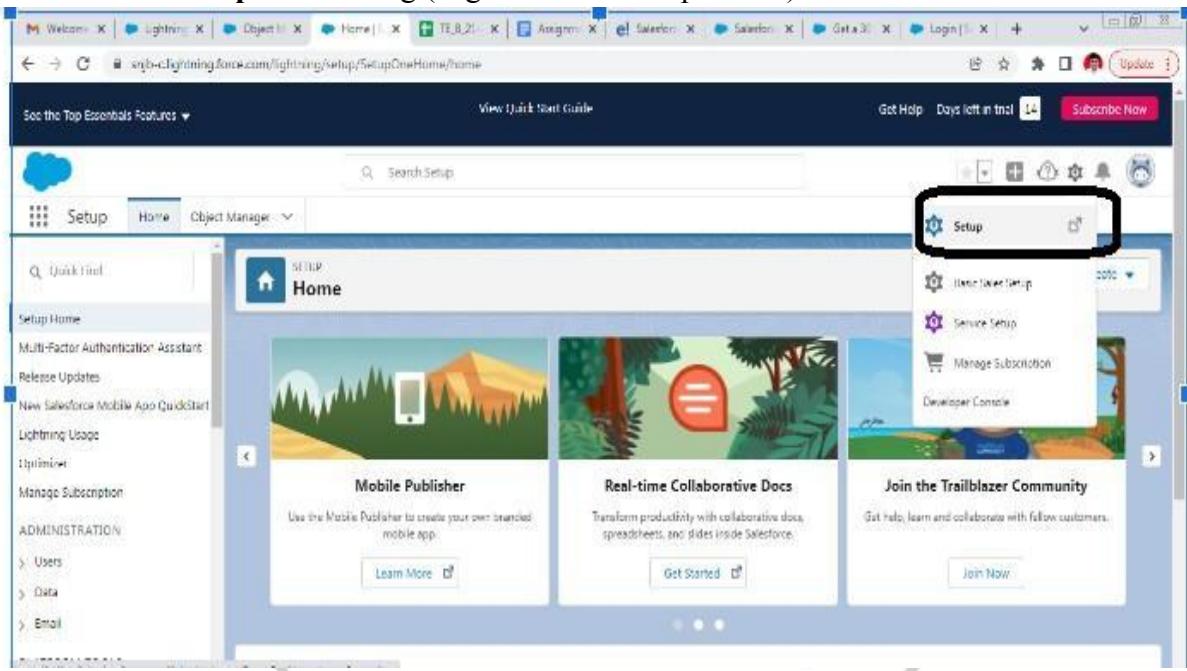


ESTD - 1928

Step-2: Logged into salesforce.com by registered user



Step-3: Click on a Setup from setting (Right side of the top corner)



ESTD - 1928

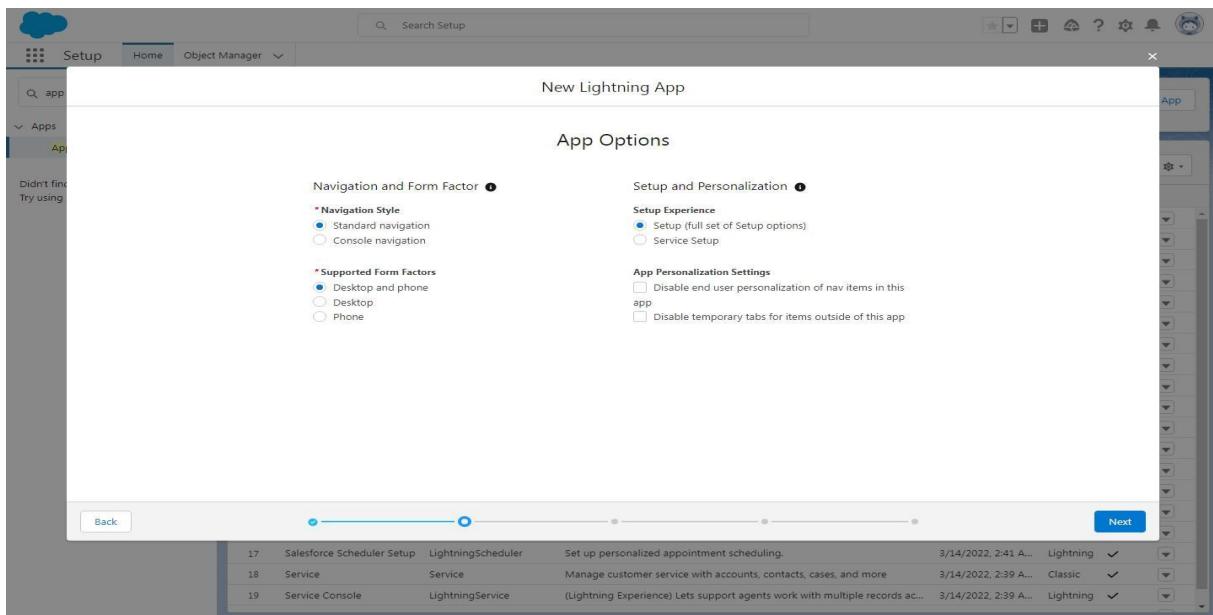
Step-4: Search for App Manager and click on New Lightning App

The screenshot shows the Salesforce Setup interface. In the top left, there's a search bar with the text "app manager". On the right, there are two prominent buttons: "New Lightning App" and "New Connected App". The main content area displays a list of 20 items under the heading "Lightning Experience App Manager". The columns include "App Name", "Developer Name", "Description", "Last Modified...", "Ap...", and "Vi...". The first item listed is "All Tabs" developed by "AllTabSet".

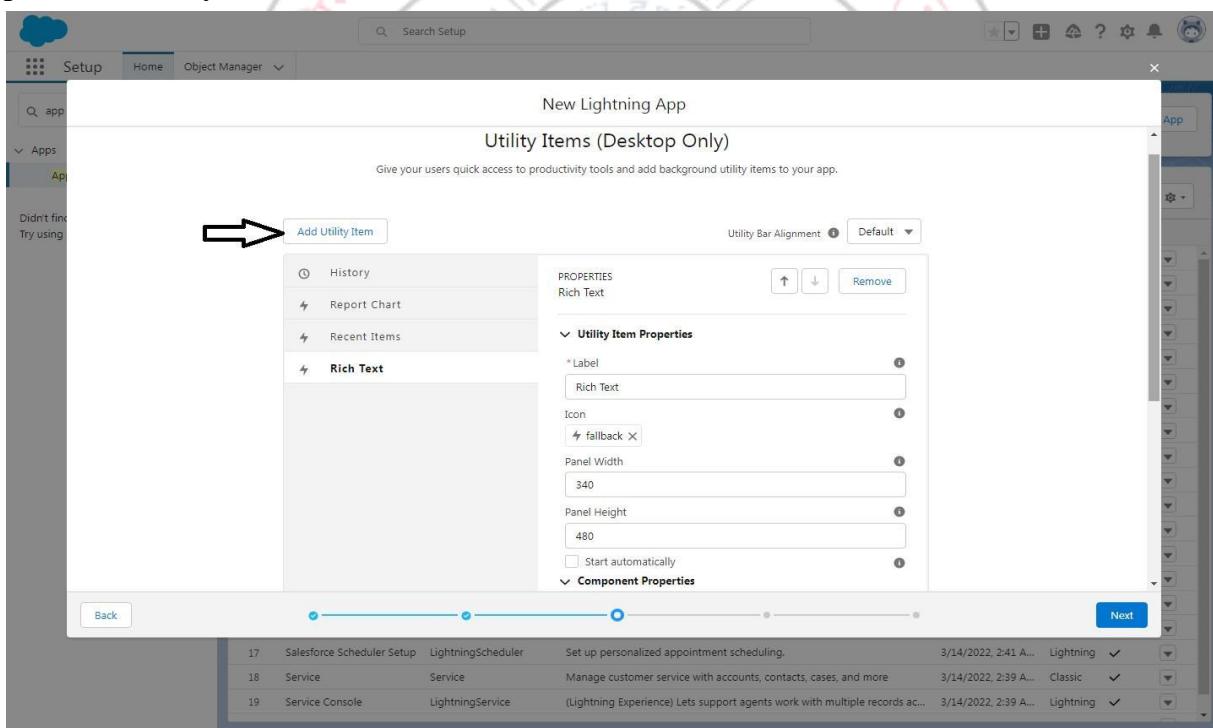
Step-5: Give App Name, fill in other details and click on Next button

This screenshot shows the "New Lightning App" configuration page. The "App Details" section contains fields for "App Name" (set to "StudentLearn"), "Developer Name" (set to "Jordy"), and "Description" (set to "This is Student Learning App"). The "App Branding" section includes a preview image of a person at a computer and a color picker set to "#0070D2". There is also an "Org Theme Options" checkbox, which is currently unchecked. At the bottom right, a "Next" button is visible.

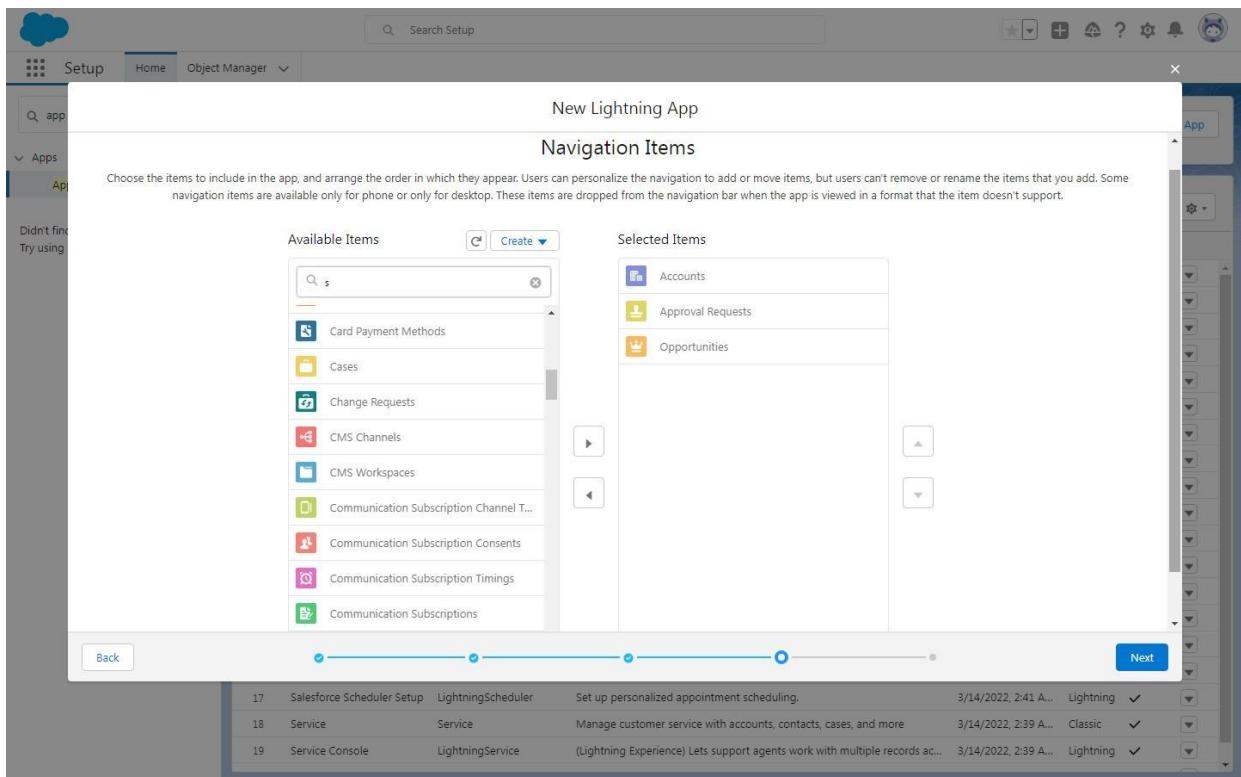
Step-6: Click Next again in-app option



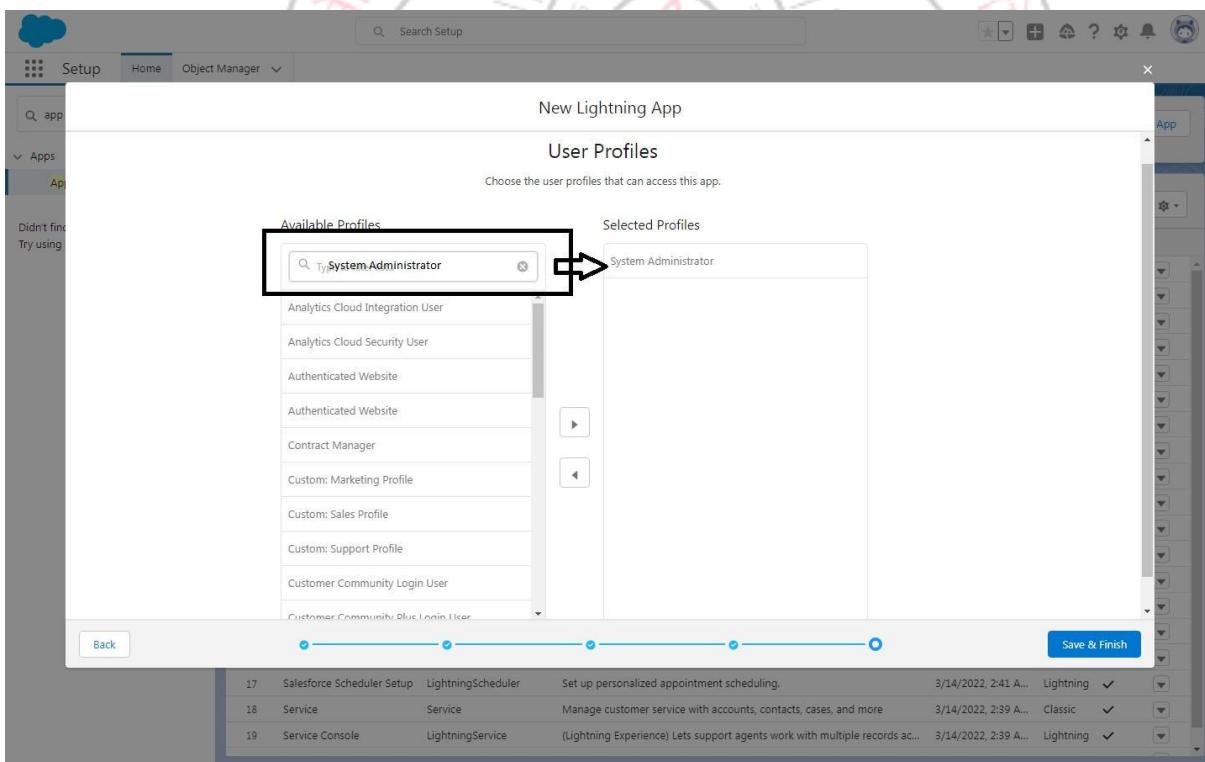
Step-7: Add Utility Items



Step-8: Add Navigation Items



Step-9: Choose the user profiles that can access this app and Click on Save & Finish button



Step-10: Search for the Application Name: StudentLearn

The screenshot shows the Salesforce Lightning Experience App Manager. At the top, there's a search bar with the text "StudentLearn". Below the search bar is a table listing various apps. The first app in the list is "All Tabs" (Developer Name: AllTabSet). The table includes columns for App Name, Developer Name, Description, Last Modified, Type, and Visibility. A red arrow points to the search bar.

App Name	Developer Name	Description	Last Modified	Type	Visibility
All Tabs	AllTabSet	Build Tableau CRM dashboards and apps	3/14/2022, 2:39 A...	Classic	✓
Analytics Studio	Insights	Build Tableau CRM dashboards and apps	3/14/2022, 2:39 A...	Classic	✓
App Launcher	AppLauncher	App Launcher tabs	3/14/2022, 2:42 A...	Lightning	✓
Bolt Solutions	LightningBolt	Discover and manage business solutions designed for your industry.	3/14/2022, 2:42 A...	Lightning	✓
Community	Community	Salesforce CRM Communities	3/14/2022, 2:39 A...	Classic	✓
Content	Content	Salesforce CRM Content	3/14/2022, 2:39 A...	Classic	✓
Digital Experiences	SalesforceCMS	Manage content and media for all of your sites.	3/14/2022, 2:39 A...	Lightning	✓
LearnerView	LearnerView	This application is for insights wrt learners	3/29/2022, 3:43 A...	Lightning	✓
Lightning Usage App	LightningInstrumentati...	View Adoption and Usage Metrics for Lightning Experience	3/14/2022, 2:39 A...	Lightning	✓
Marketing	Marketing	Best-in-class on-demand marketing automation	3/14/2022, 2:39 A...	Classic	✓
Platform	Platform	The fundamental Lightning Platform	3/14/2022, 2:39 A...	Classic	✓
Queue Management	QueueManagement	Create and manage queues for your business.	3/14/2022, 2:39 A...	Lightning	✓
Sales	Sales	The world's most popular sales force automation (SFA) solution	3/14/2022, 2:39 A...	Classic	✓
Sales	LightningSales	Manage your sales process with accounts, leads, opportunities, and more	3/14/2022, 2:39 A...	Lightning	✓
Sales Console	LightningSalesConsole	(Lightning Experience) Lets sales reps work with multiple records on one ...	3/14/2022, 2:39 A...	Lightning	✓
Salesforce Chatter	Chatter	The Salesforce Chatter social network, including profiles and feeds	3/14/2022, 2:39 A...	Classic	✓
Salesforce Scheduler Setup	LightningScheduler	Set up personalized appointment scheduling.	3/14/2022, 2:41 A...	Lightning	✓
Service	Service	Manage customer service with accounts, contacts, cases, and more	3/14/2022, 2:39 A...	Classic	✓
Service Console	LightningService	(Lightning Experience) Lets support agents work with multiple records ac...	3/14/2022, 2:39 A...	Lightning	✓

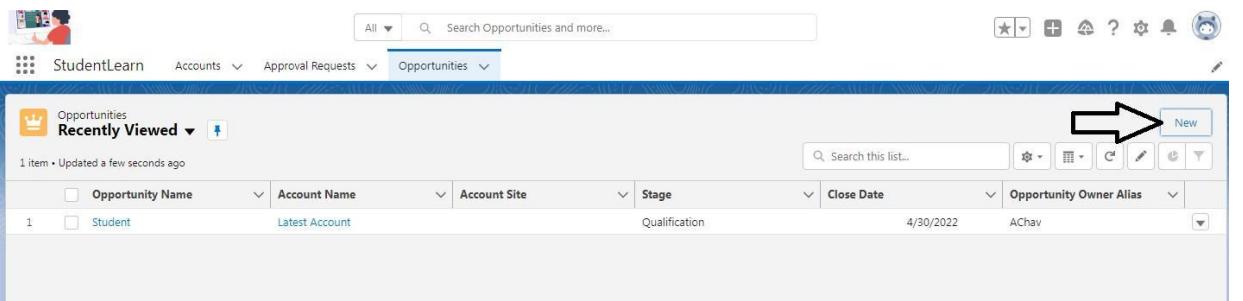
Step-11: Finally, we get the Custom Fields / Labels as shown below,

The screenshot shows the Salesforce Accounts page. At the top, there's a search bar with the placeholder "Search Accounts and more...". Below the search bar is a table listing accounts. The first account in the list is "Latest Account" (Account Name: Latest Account, Account Owner Alias: AChav). A red arrow points to the bottom of the page where the text "Custom Fields / Labels" is displayed, followed by a large downward-pointing arrow.

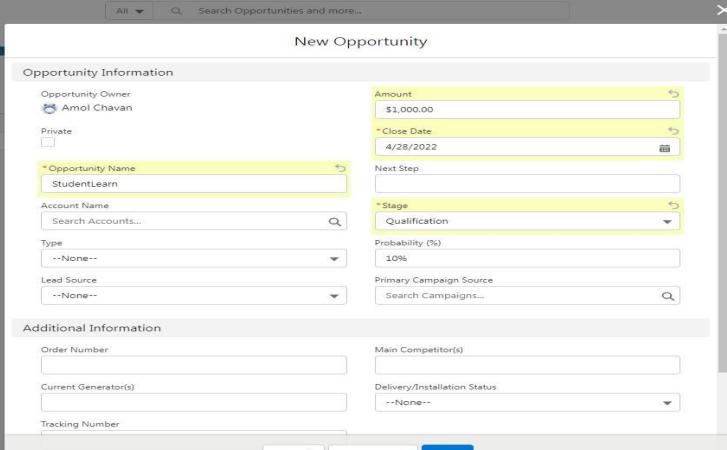
Account Name	Account Site	Phone	Account Owner Alias
Latest Account			AChav

Custom Fields / Labels

Step- 12: Click on New Button for Adding Records & Finish

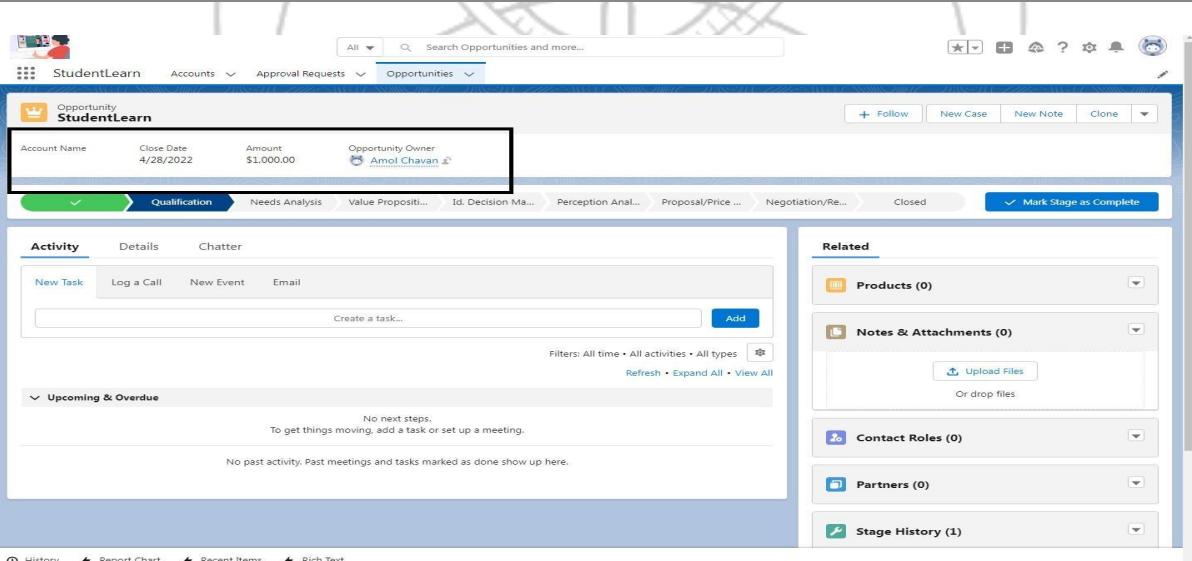


The screenshot shows the Salesforce Opportunities list view. A black arrow points to the 'New' button in the top right corner of the list header.



The screenshot shows the 'New Opportunity' dialog box. It contains the following sample data:

Field	Value
Opportunity Name	StudentLearn
Account Name	Latest Account
Stage	Qualification
Close Date	4/28/2022
Opportunity Owner Alias	AChav
Amount	\$1,000.00
Probability (%)	10%



The screenshot shows the Opportunity record detail page for 'StudentLearn'. The record summary at the top includes:

- Account Name: StudentLearn
- Close Date: 4/28/2022
- Amount: \$1,000.00
- Opportunity Owner: Amol Chavan

The 'Activity' tab is selected, showing a 'New Task' button and a list of tasks. The 'Related' tab shows sections for Products, Notes & Attachments, Contact Roles, Partners, and Stage History.

Frequently Asked Questions:

- i) What is Lightning Experience?
- ii) What Object Manager?
- iii) What are Custom Fields?
- iv) What is App Manager?
- v) What is User Profile in Ligtng App?

Conclusion:

Hence, successfully created the custom Application SalesForce.com.



ESTD - 1928