

Dr. AMBEDKAR INSTITUTE OF TECHNOLOGY

Near Jnana Bharathi Campus, Bengaluru-560 056.

(An Autonomous Institution, Aided by Government of Karnataka)



LAB REPORT

Submitted in the Partial Fulfillment of the Cloud Programming Laboratory

18CSL77

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

SUBMITTED

BY

SHASHANK M KADIWAL

1DA18CS141

Department of Computer Science & Engineering

2021-22

CONTENTS

<u>Chapter No</u>	<u>Title</u>	<u>Page No.</u>
	Part A	
Chapter 1	Salesforce	1
Chapter 2	CloudAnalyst Simulations	106
Chapter 3	Google App Engine	112
Chapter 4	Amazon Web Services - RDS	121
Chapter 5	VM Ware Workstation	137
	File Transfer in Virtual Machines	160
	Part B	
Chapter 6	Mini Project	181
Chapter 7	Industry Certifications - Salesforce	

SALESFORCE

Salesforce is an American cloud-based software company headquartered in San Francisco, California. It provides customer relationship management (CRM) service and also provides enterprise applications focused on customer service, marketing automation, analytics, and application development.

History:

The company was founded on February 3, 1999 by former Oracle executive Marc Benioff, together with Parker Harris, Dave Hoelterhoff, and Frank Dominguez as a software as a service (SaaS) company, and was launched publicly between September and November 1999.

In June 2004, the company had its initial public offering on the New York Stock Exchange under the stock symbol CRM and raised US\$110 million. Early investors include Larry Ellison, Magdalena Yesil, Halsey Minor, Stewart Henderson, Mark Iscaro, and Igor Sill, a founding member of Geneva Venture Partners.

In October 2014, Salesforce announced the development of its Customer Success Platform to tie together Salesforce's services, including sales, service, marketing, analytics, community, and mobile apps. In October 2017, Salesforce launched a Facebook Analytics tool for business-to-business marketers. In September 2018, Salesforce partnered with Apple intended on improving apps for businesses.

In February 2020, co-chief executive officer Keith Block stepped down from his position in the company. Marc Benioff remained as chairman and chief executive officer.

On December 1, 2020, it was announced that Salesforce would acquire Slack for \$27.7 billion. The acquisition closed on July 21.

In February 2021, Salesforce announced that CFO Mark Hawkins would be retiring from his position after six years of working for the company; however, retaining a position as CFO emeritus until October. Amy Weaver was selected as his replacement.

Salesforce.com's customer relationship management (CRM) service comprises several broad categories: Commerce Cloud, Sales Cloud, Service Cloud, Data Cloud (including Jigsaw), Marketing Cloud, Community Cloud (including Chatter), Manufacturing Cloud, Analytics Cloud, App Cloud, Vaccine Cloud,IoT and Work.com with over 100,000 customers.

Main services

Salesforce's main services are tools for case, task and issue management. It also gives customers tracking abilities for their raised cases and conversation features for social networking Web sites, provides analytical tools and other services including email alert, Google search, and access to customers' entitlement and contracts. They also partner with companies like IBM, Accenture, and Saggezza to help integrate Salesforce's cloud-based services into their businesses.

Lightning Platform

Lightning Platform (also known as Force.com) is a platform as a service (PaaS) that allows developers to create add-on applications that integrate into the main Salesforce.com application. [failed verification] These third-party applications are hosted on Salesforce.com's infrastructure.

Force.com applications are built using declarative tools, backed by Lightning [further explanation needed] and Apex, a proprietary Java-like programming language for Force.com, as well as Visualforce, a framework including an XMLsyntax typically used to generate HTML. The Force.com platform typically receives three complete releases a year. As the platform is provided as a service to its developers, every single development instance also receives all these updates.

In 2015, a new framework for building user interfaces – Lightning Components – was introduced in beta. Lightning components are built using the open-source Aura Framework but with support for Apex as the server-side language instead of Aura's JavaScript dependency. This has been described as an alternative to, not necessarily a replacement for, Visualforce pages.

As of 2013, the Force.com platform has 1.4 million registered developers. Lightning Base Components is the component library built on top of Lightning Web Components.

Experience Cloud

Experience Cloud (formerly Community Cloud) provides Salesforce customers the ability to create online web properties for external collaboration, customer service, channel sales, and other custom portals in their instance of Salesforce. Tightly integrated to Sales Cloud, Service Cloud, and App Cloud, Experience Cloud can be quickly customized to provide a wide variety of web properties. Experience Cloud combines the functionality of the former Salesforce Customer and Partner Portals with some additional features.

Work.com

Work.com, previously Rypple, is a social performance management platform for managers and employees. It allows continuous coaching, real-time feedback, and recognition. It is aimed at sales management, customer service, marketing, and can be utilised by human resource departments.

Work.com, then known as "Rypple", was founded by Daniel Debow and David Stein, to create a simple way of asking for feedback anonymously at work. The company was formed in May 2008 and their client list included Mozilla, Facebook, LinkedIn and the Gilt Groupe. Rypple aims to get employees to build and manage their own coaching networks.

In September 2011, Rypple announced that they had hired Bohdan Zabawskyj as its Chief Technology Officer. In 2011, Rypple developed a more formalized management methodology called OKR ("Objectives and Key Results") for Spotify. Rypple also partnered with Facebook to create "Loops", short for "feedback loops", which gathers feedback from co-workers, including praise, progress against goals, and coaching from supervisors into one channel.

In December 2011, Salesforce.com announced that they would acquire Rypple. The transaction was completed in 2012 and Rypple was rebranded as Work.com in September 2012.

AppExchange

Launched in 2005, the Salesforce AppExchange is an online application marketplace for third-party applications that run on the Force.com platform. Applications are available for free, as well as via yearly or monthly subscription models. Applications available range from integrations with SharePoint to mobile approval management. As of June 2016, it features 2,948 applications which have driven 3+ million installs. The "AppExchange" is also a place customers can search for cloud consulting partners to help them implement the technology in their own organization. Cloud consulting partners for Salesforce include large companies like IBM's "Bluewolf" and Accenture as well as smaller ones like Cloureach.

myTrailhead

Launched in 2019, Salesforce's myTrailhead is an online training platform that can be customized for the specific needs of its customers. The platform extends functionality to provide users with training content specific to their usage of Salesforce and enables them to create and publish their own training content and programs.



Technologies:

Salesforce is powered by the Model–view–controller architecture.

Apex

Apex is a proprietary programming language provided by the Force.com platform to developers similar to Java and C#. It is a strongly typed, object-oriented, case-insensitive programming language, following a dot-notation and curly-brackets syntax. Apex can be used to execute programmed functions during most processes on the Force.com platform including custom buttons and links, event handlers on record insertion, update, or deletion, via scheduling, or via the custom controllers of Visualforce or Lightning Experience pages.

Due to the multitenant nature of the platform, the language has strictly imposed governor limitations to guard against any code monopolizing shared resources. Salesforce provides a series of asynchronous processing methods for Apex to allow developers to produce longer-running and more complex Apex code.

Lightning

In 2014, Salesforce made public the front end of its platform, called Lightning. This component-based framework is what the Salesforce mobile app is built on. Salesforce built on this framework in 2015 by releasing the Lightning Design System, an HTML style framework with default CSS styling built in. This framework allows customers to build their own components to either use in their internal instances or sell on the AppExchange.

The Salesforce Lightning App Builder is a tool for rapid application development of responsive web interfaces. This interface allows for different screens to be put together based on Lightning components. This can be used as layouts for records or specific applications.

Lightning Experience, released in 2016, is the new redesigned interface in Salesforce for processes enhancement. Since then all the apps available on AppExchange need to be Lightning and those built on Classic have to migrate to Lightning as Classic is not to be updated any more by Salesforce. The platform

offers an option for developers to employ migration techniques to enable the new user-friendly interface and switch to Lightning.

What is Salesforce used for?

- Engage customers with relevant, empathetic digital marketing from anywhere.
- Sell smarter and grow your business faster from anywhere.
- Quickly launch and scale ecommerce built around your customer — from anywhere.
- Provide great customer service from anywhere.
- Go digital fast and empower your teams to work from anywhere.



1. Create a web application to enter student details like Name, USN, Semester, Section and CGPA to a database on Salesforce Cloud Platform

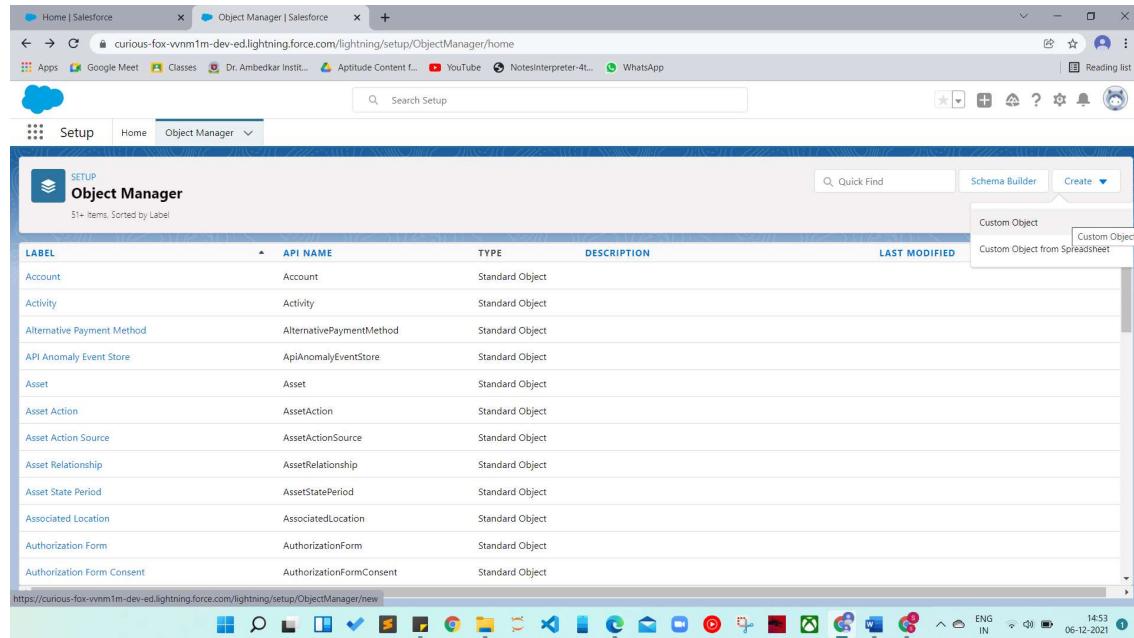
Steps:

1. Launch your Salesforce Trailhead Playground by opening any module and Switch to

Lightning Experience if you are currently in Salesforce Classic by clicking your picture in the

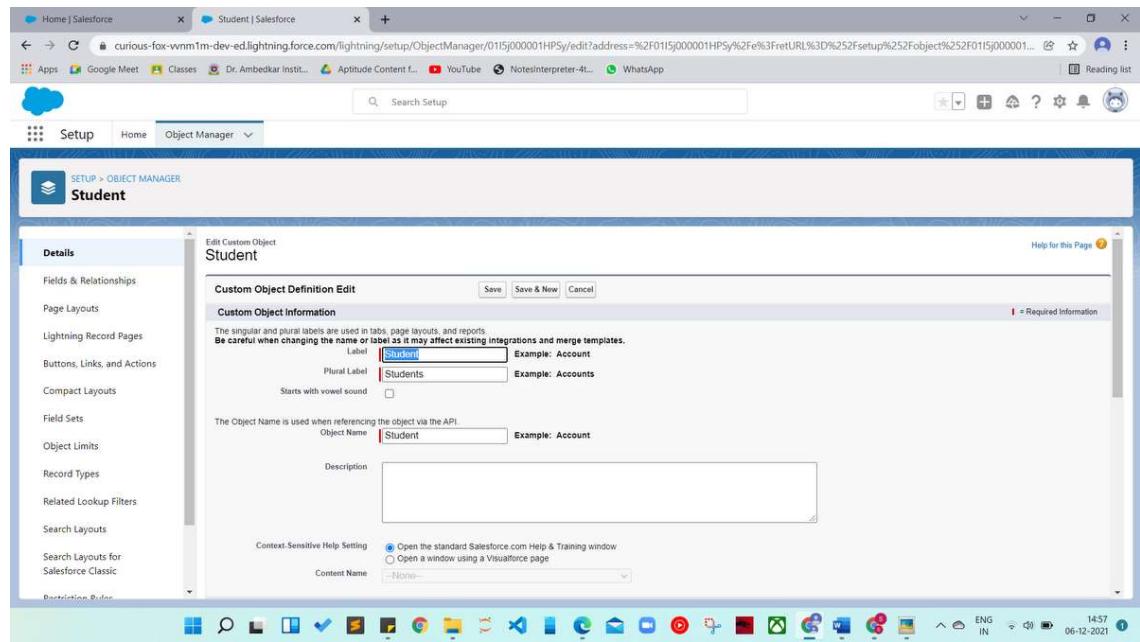
right top corner and then click on “Switch to Lightning Experience”.

2. Then go to Setup gear icon and click “Setup”.
3. Click on “Object Manager” and click “Create> Custom Object” to create new Custom Object.



The screenshot shows the Salesforce Object Manager page. At the top, there's a navigation bar with tabs for 'Setup' and 'Object Manager'. Below the navigation is a search bar labeled 'Search Setup'. On the right side of the page, there are buttons for 'Custom Object' and 'Custom Object from Spreadsheet'. The main area displays a table of objects, with columns for 'LABEL', 'API NAME', 'TYPE', 'DESCRIPTION', and 'LAST MODIFIED'. The table lists various standard objects such as Account, Activity, Alternative Payment Method, API Anomaly Event Store, Asset, Asset Action, Asset Action Source, Asset Relationship, Asset State Period, Associated Location, Authorization Form, and Authorization Form Consent. The 'LAST MODIFIED' column shows the last update date for each object.

4. Name the object “Student”.
5. Allow Reports and Allow Search.
6. Check the box in front of “Launch New Custom Tab Wizard after saving this custom object”.



7. To create a Tab for the Object: Select any Tab Style for the object “Student”.

Click Next, Next, leave the defaults and save.

8. To add fields to the Object: Go to “Fields & Relationships” option of Student object and Click “New”.

9. Add the following fields one after the other:

- Field Label: USN (Length 10), Data Type: Text, provide an example USN as Help Text, make it as Required Field and Don’t allow Duplicate Values and make it as Case Insensitive.
- Field Label: Section (Length 1), Data Type: Text, make it as Required Field.
- Field Label: Semester (Length 1, Decimal Place 0), Data Type: Number, make it as Required Field.
- Field Label: CGPA (Length 2, Decimal Places 2), Data Type: Number, make it as Required Field.

The screenshot shows the Salesforce Object Manager interface for the 'Student' object. The left sidebar lists various setup options like Page Layouts, Lightning Record Pages, Buttons, Links, and Actions. The main area displays the 'Fields & Relationships' section with the following details:

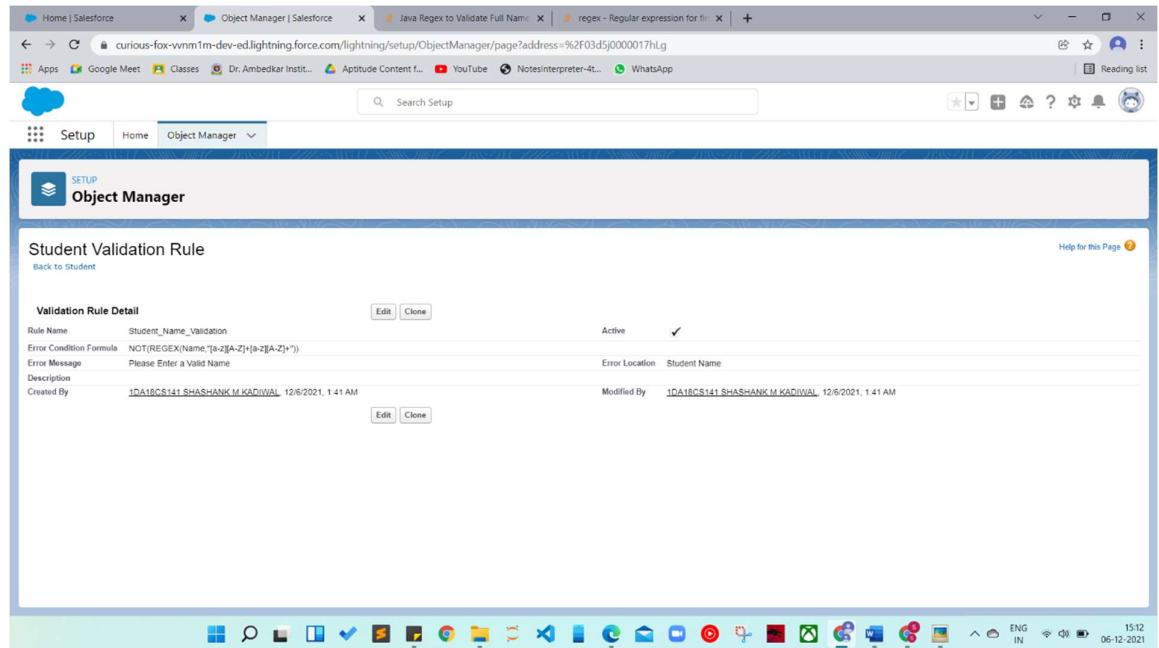
FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
CGPA	CGPA_c	Number(2, 2)		
Created By	CreatedById	Lookup(User)		
Last Modified By	LastModifiedById	Lookup(User)		
Owner	OwnerId	Lookup(User, Group)		✓
Section	Section_c	Text(1)		
Semester	Semester_c	Number(1, 0)		
Student Name	Name	Text(80)		✓
USN	USN_c	Text(10) (Unique Case Sensitive)		✓

Validation Rules:

1. Name validation:

To add a rule to the Student Name so that the name should only start with letter but not digit:

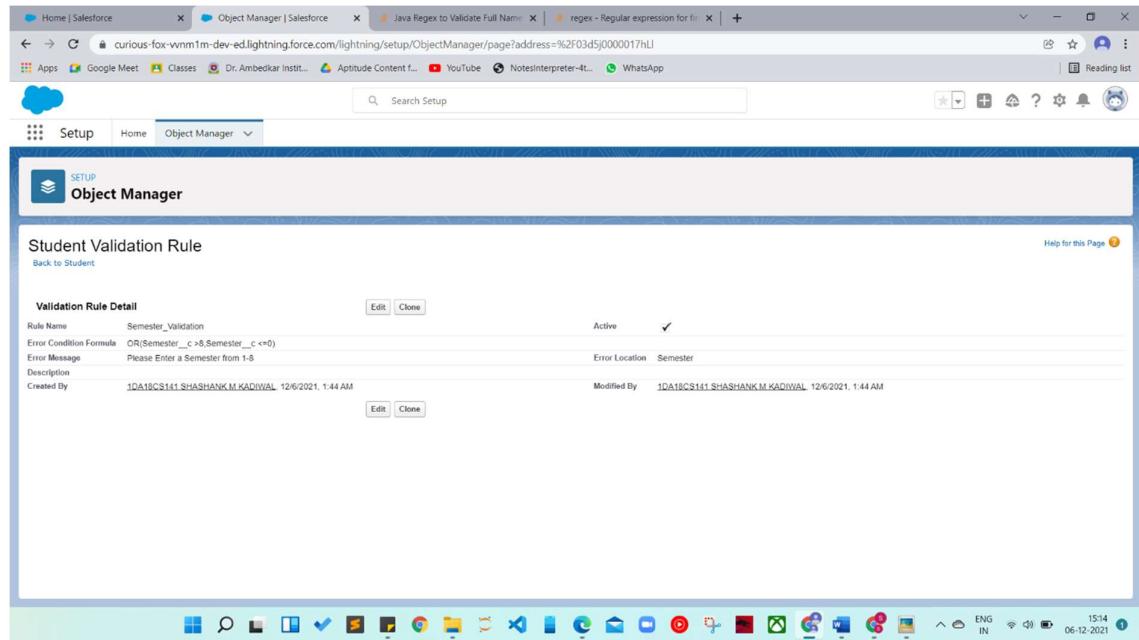
- Go to Validation Rule of Student Object and click “New”
- Name it as “Student Name Validation”
- Error Condition Formula: “NOT (REGEX (Name, "[a-zA-Z][a-zA-Z]+"))”
- Error Message: “Please Enter a valid name”.
- Error Location: Field –Student Name.
- Click Save.



2. Semester Validation:

To add a rule to the Semester so that it should always be greater than 0 and less than or equal to 8:

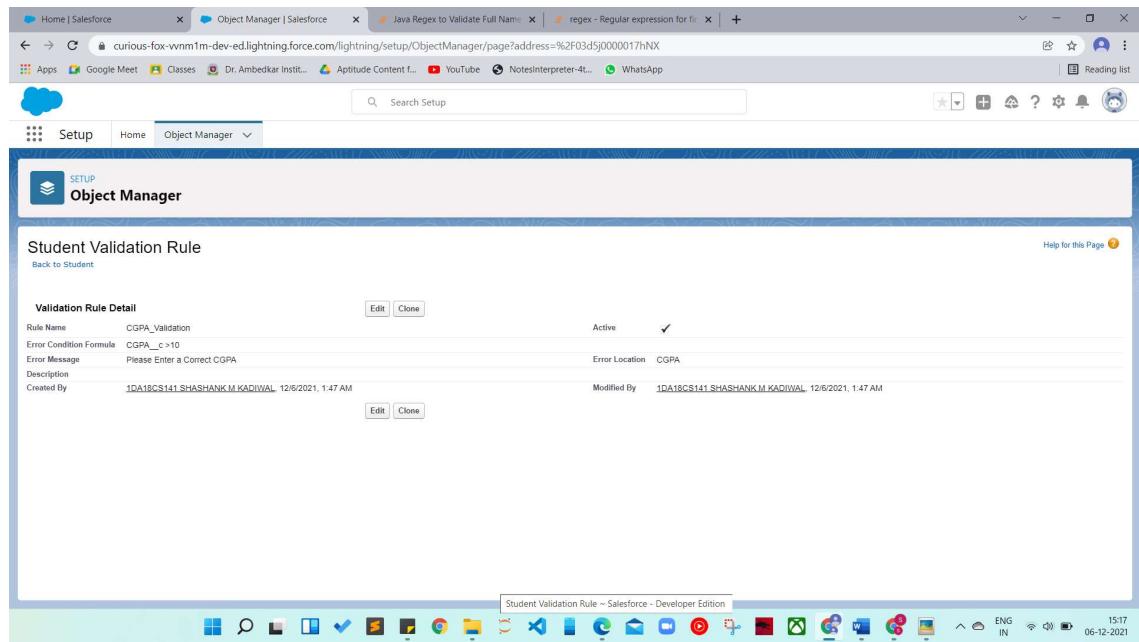
- Go to Validation Rule of Student Object and click “New”.
- Name it as “Semester validation”.
- Error Condition Formula: OR (Semester__c >8, Semester__c <=0).
- Error Message: Please Enter a Semester from 1-8.
- Error Location: Field – Semester.
- Click Save.



3.CGPA Validation:

To add a rule to the CGPA so that it should not take CGPA greater than 10:

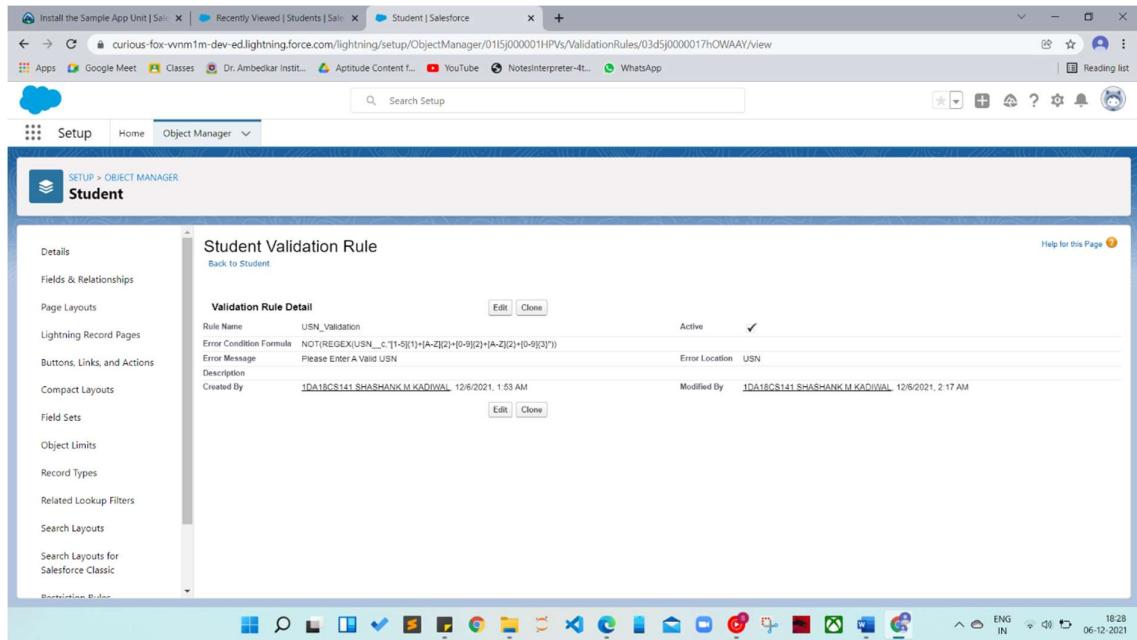
- Go to Validation Rule of Student Object and click “New”
- Name it as “CGPA validation”.
- Error Condition Formula: CGPA__c > 10
- Error Message: Please Enter a Correct CGPA.
- Error Location: Field –CGPA
- Click Save



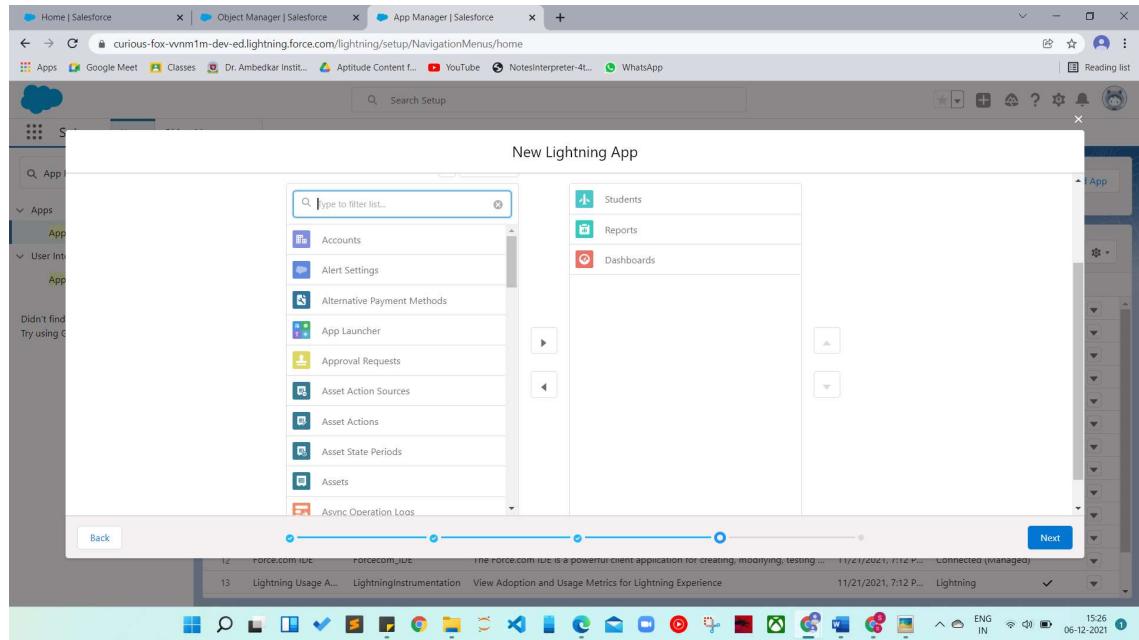
4.USN Validation:

To add a rule to the USN so that it should validate only student's serial number.

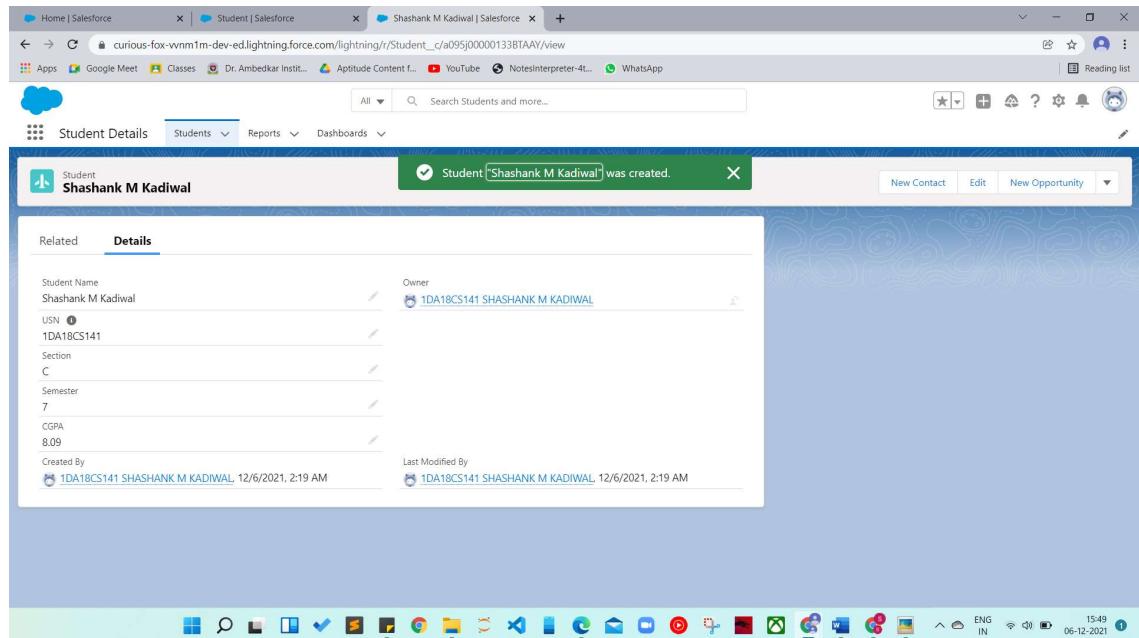
- Go to Validation Rule of Flight Object and click “New”
- Name it as “USN Validation”.
- Error Condition Formula: NOT(REGEX(USN_c,"[1-5]{1}+[A-Z]{2}+[0-9]{2}+[A-Z]{2}+[0-9]{3}))
- Error Message: Please enter a valid USN.
- Error Location: Field – USN
- Click Save.



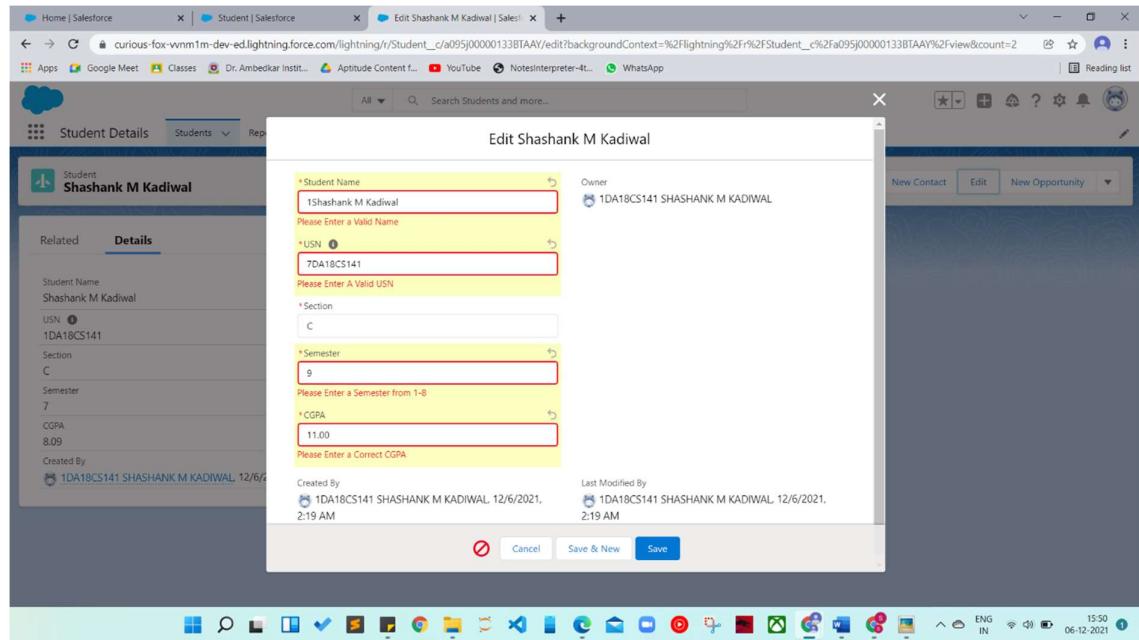
10. To create an application: Go to “Setup” and type “App Manager” in Quick Find Box.
11. Click on “New Lightning App” to create a Lightning Application.
12. Name it as “Student Details”, give the description for your application.
13. Uploading Image and changing colours are optional, then click Next.
14. Navigation Style: Standard Navigation, click Next.
15. No need to add any Utility Bar, click Next.
16. Add the following Items: Students, Reports and Dashboards, click Next.
17. Assign it to System Administrator Profile by selecting System Administrator and pressing right arrow and then click Save & Finish.



18. Go to App Manager, select your application and select Students and click “New” to add some details to your application.



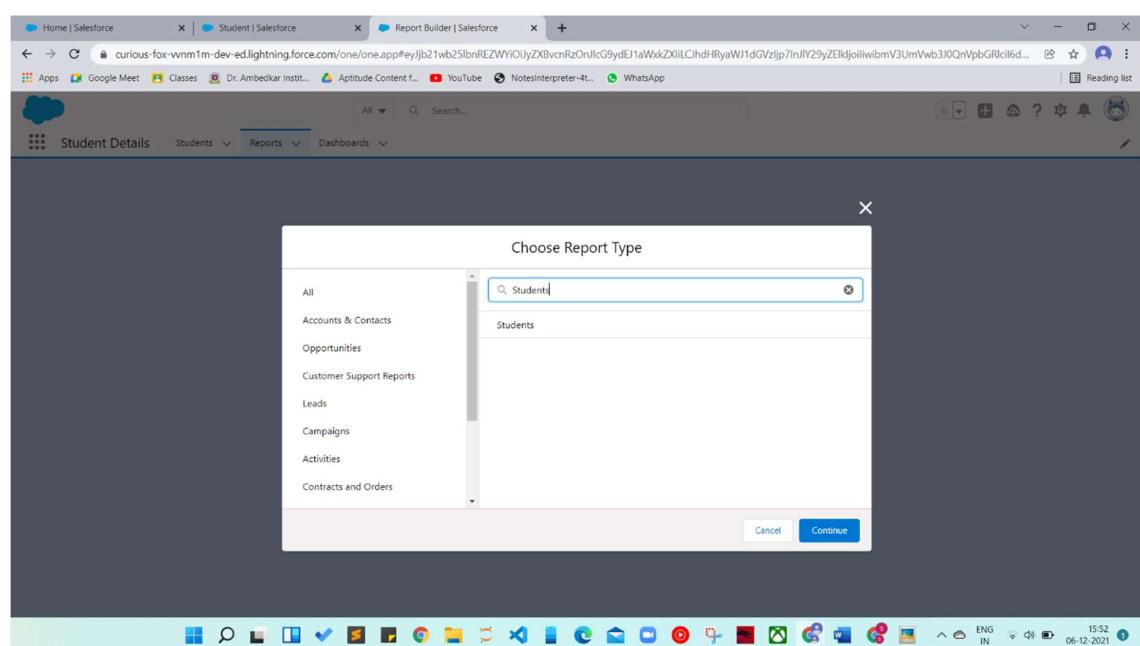
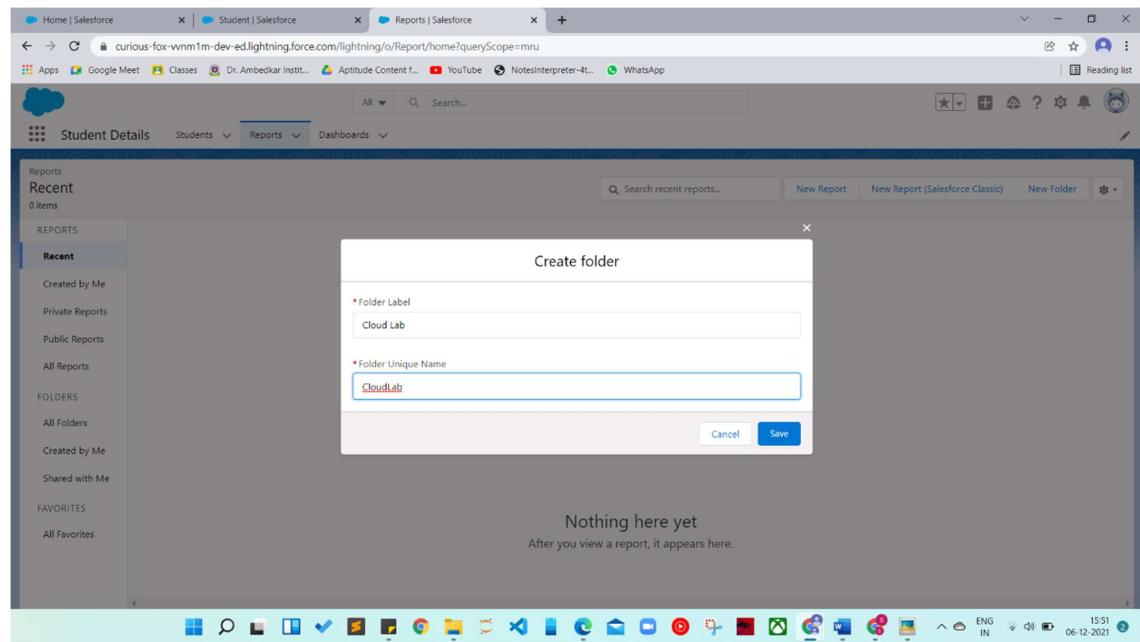
Make sure you will get error messages when you give invalid Name, USN, Semester and CGPA.



Reports and Dashboards:

To Create a Students Report:

- Go to “Reports tab” Click on “New Folder” And give it any name and Click Save.
- Click on “New Report” and from search bar Search for “Students” and then select it and then click continue.
- Add the required Columns to get the Completed Entered data.
- If you Want the report to be grouped by any specific Fields then Search for the field in “Add groups” otherwise it is optional.
- Click on save and name the report as “New Students Report” and then select the folder which you have created.
- Click Save and then Click Run

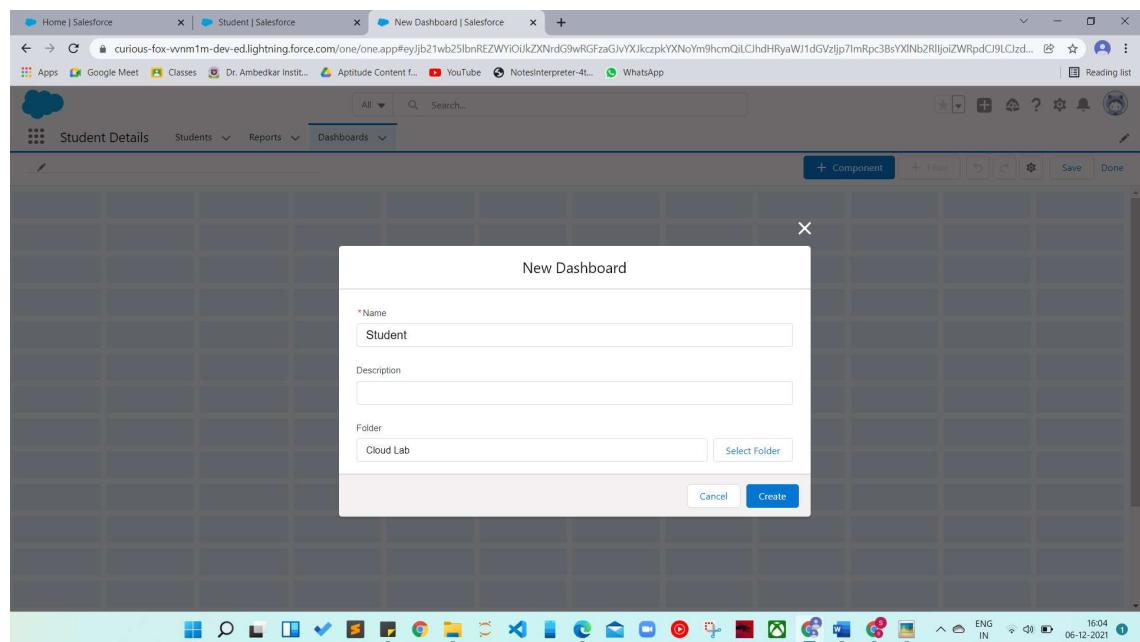
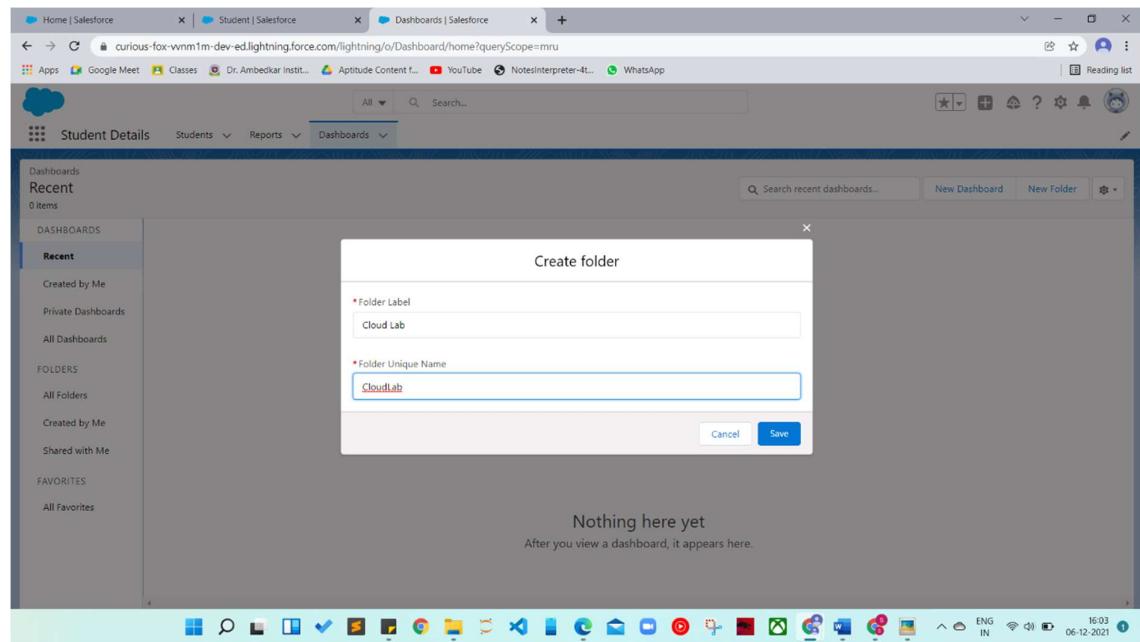


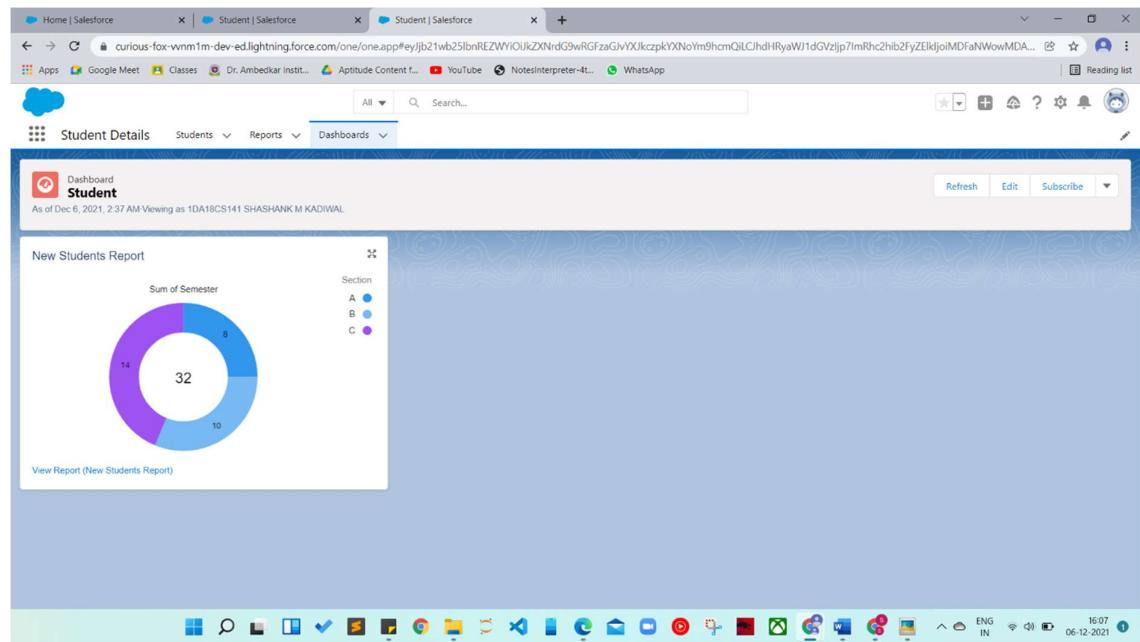
The screenshot shows the Salesforce Report Builder interface. The report is titled "New Students Report" and is set to run against the "Students" object. The report structure is defined with "Groups" (Section A, B, C) and "Columns" (Student Name, Semester, CGPA, USN). The data preview shows 10 records:

Section	Student: Student Name	Semester	CGPA	USN
A (2)	Saiteja	5	9.50	1DA19CS137
	Advaith	3	9.12	1DA20CS007
B (2)	Shrivatsa	5	8.65	1DA19CS150
	Dhanush S	5	8.86	1DA19CS129
C (2)	Sanjay Revanna	7	8.90	1DA18CS137
	Shashank M Kadiwal	7	8.09	1DA18CS141

To Create a Students Dashboard:

- Go to “Dashboard tab” and then click on “New Folder” and give it any Name.
- Click on “New Dashboard” and then name it as “Students Dashboard” and select folder that you have created Click on Create.
- Click on the report that you have created and click on that and click select.
- Select any style to represent the data in dashboard.
- Add any filter(s), otherwise it is optional.
- Click on Save and Click Run.

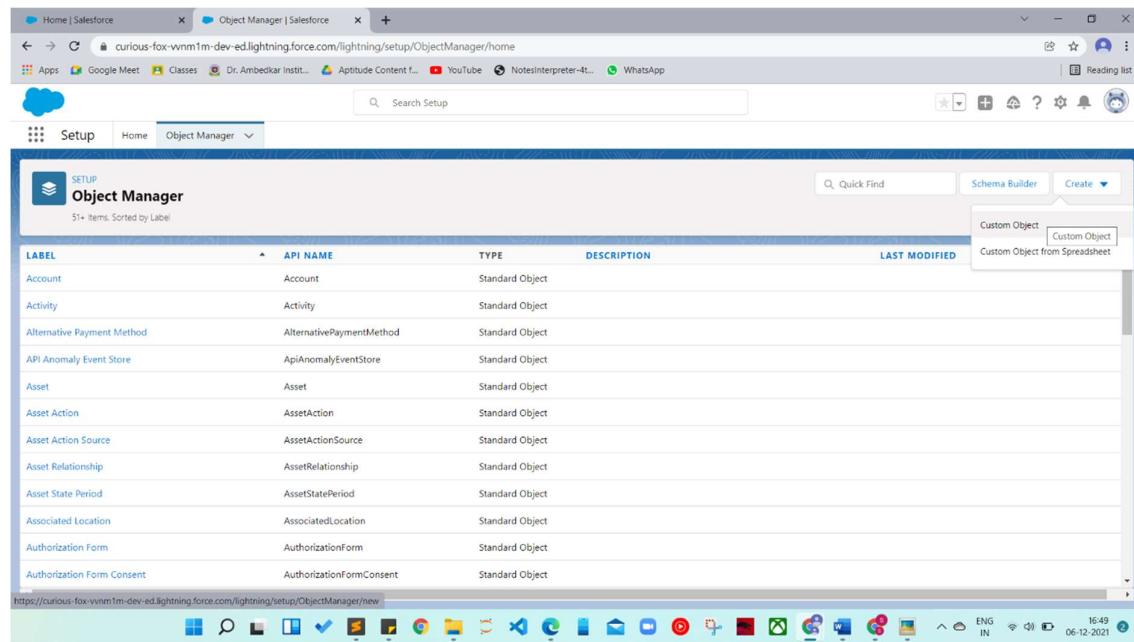




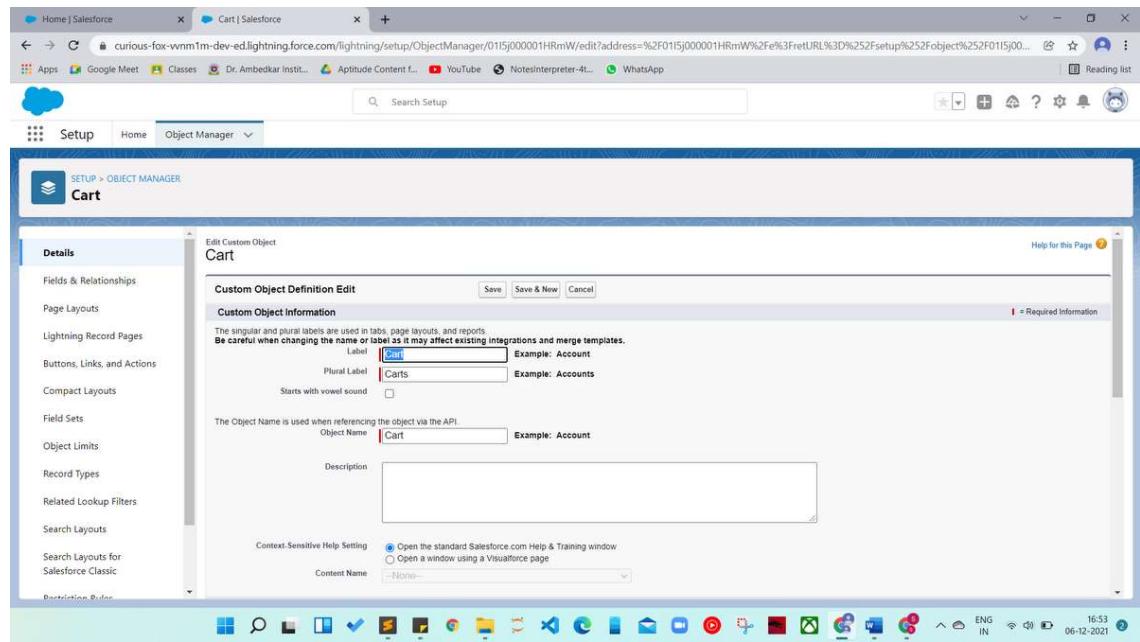
2. Create a Web Application to implement an online cart for adding items to a shopping cart and deleting it.

STEPS:

1. Launch your Salesforce Trailhead Playground by opening any module and Switch to Lightning Experience if you are currently in Salesforce Classic by clicking your picture in the right top corner and then click on “Switch to Lightning Experience”.
2. Then go to Setup gear icon and click “Setup”.
3. Click on “Object Manager” and click “Create> Custom Object” to create new Custom Object.



4. Name the object “Cart”.
5. Allow Reports and Allow Search.
6. Check the box in front of “Launch New Custom Tab Wizard after saving this custom object”
7. To create a Tab for the Object: Select any Tab Style for the object “Cart”. Click Next, Next, leave the defaults and save.



8. To add fields to the Object: Go to “Fields & Relationships” option of cart object and Click “New”.

9. Add the following fields one after the other:

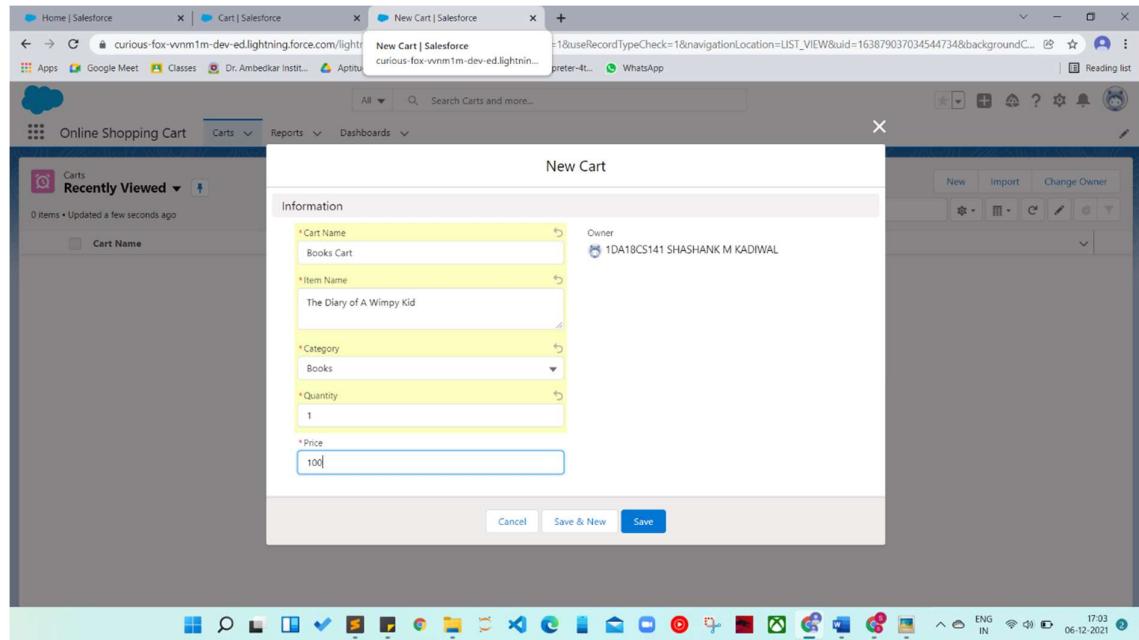
- Field Label: Item Name, Data Type: Text Area, make it as Required Field.
- Field Label: Category, Data Type: Picklist, click radio button in front of Enter values, with each value separated by a new line value are: Books, Electronics & Accessories, Furniture & Home Appliances, Fashion – Men, Fashion – Women, Fashion – Kids, Footwear and Others.
- Make it as Required Field and Restrict the values to the values in the picklist.
- Field Label: Quantity, Data Type: Number, make it as Required Field.
- Field Label: Price, Data Type: Currency (Length 16, Decimal Places 2), Make it as Required Field.

The screenshot shows the Salesforce Setup interface with the 'Object Manager' for the 'Cart' object. The 'Fields & Relationships' tab is selected. The table lists the following fields:

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Cart Name	Name	Text(60)		✓
Category	Category__c	Picklist		✗
Created By	CreatedById	Lookup(User)		✗
Item Name	Item_Name__c	Text Area(255)		✗
Last Modified By	LastModifiedById	Lookup(User)		✗
Owner	OwnerId	Lookup(User,Group)		✓
Price	Price__c	Currency(16, 2)		✗
Quantity	Quantity__c	Number(18, 0)		✗

To create an application:

10. Go to “Setup” and type “App Manager” in Quick Find Box.
11. Click on “New Lightning App” to create a Lightning Application.
12. Name it as “Online Shopping Cart”, give the description for your application.
13. Uploading Image and changing colours are optional, then click Next.
14. Navigation Style: Standard Navigation, click Next.
15. No need to add any Utility Bar, click Next.
16. Add the following Items: Carts, Reports and Dashboards, click Next.
17. Assign it to System Administrator Profile by selecting System Administrator and pressing right arrow and then click Save & Finish.
18. Go to App Manager, select your application and select Carts and click “New” to add some details to your application.



Reports and Dashboards:

To Create a Carts Report:

- Go to “Reports tab” Click on “New Folder” And give it any name and then click Save.
- Click on “New Report” and from search bar Search for “Carts” and then select it and click Continue. • Add the required Columns to get the Completed Entered data.
- If you Want the report to be grouped by any specific Fields then Search for the field in “Add groups” otherwise it is optional.
- Click on save and name the report as “New Carts Report” and then select the folder which you have created.
- Click Save and then Click Run

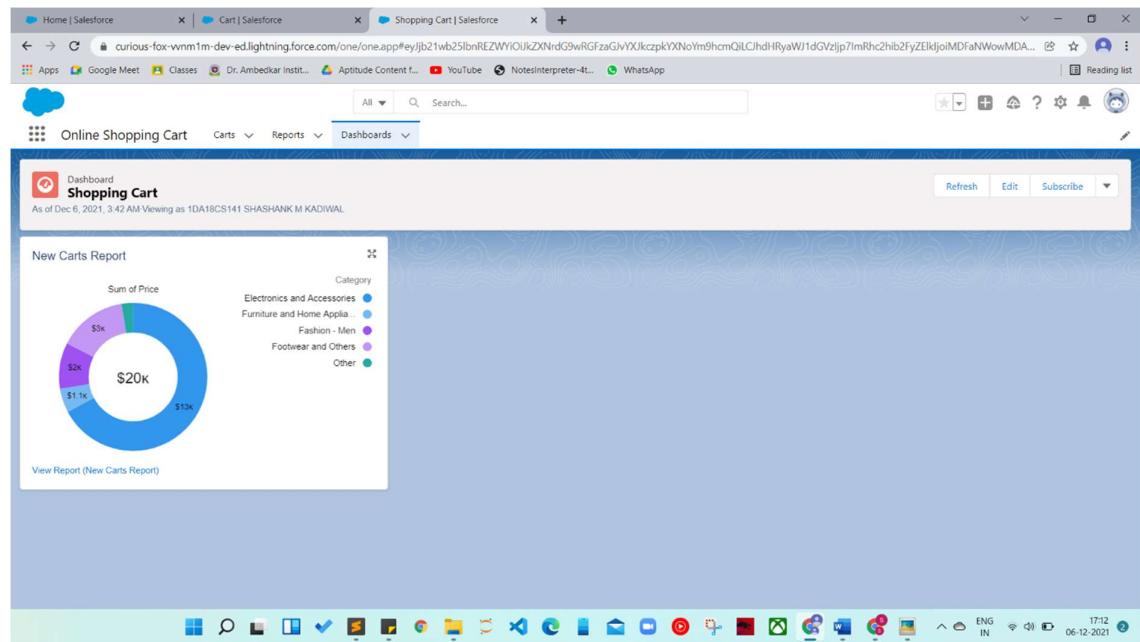
The screenshot shows a Salesforce interface with three tabs at the top: 'Home | Salesforce', 'Cart | Salesforce', and 'New Carts Report | Salesforce'. The current view is 'New Carts Report'. The page title is 'Report: Carts New Carts Report'. It displays a table with the following data:

Total Records	Total Price	Total Quantity		
8	\$19,950.00	16		
<input type="checkbox"/> Category ↑	<input type="checkbox"/> Cart: Cart Name ↓	<input type="checkbox"/> Item Name ↓	<input type="checkbox"/> Price ↓	<input type="checkbox"/> Quantity ↓
<input type="checkbox"/> Books (2)	Books Cart	The Diary of A Wimpy Kid	\$100.00	1
	Fiction	Harry Potter	\$400.00	2
<input type="checkbox"/> Electronics and Accessories (2)	Electronics Cart	Mi Redmi Note 4	\$11,000.00	1
	Headphones	Sony Bluetooth Headphones	\$2,400.00	1
<input type="checkbox"/> Furniture and Home Appliances (2)	Wall Decorations	Wall Stickers	\$400.00	5
	Home Decors	Buddha Idol	\$650.00	2
<input type="checkbox"/> Fashion - Men (1)	Men Fashion	Lee Cooper Jeans	\$2,000.00	3
<input type="checkbox"/> Footwear and Others (1)	Men Shoes	Benetton Casual Shoes	\$3,000.00	1
Total (8)			\$19,950.00	16

At the bottom, there are checkboxes for 'Row Counts', 'Detail Rows', 'Subtotals', and 'Grand Total'. Below the table is a toolbar with various icons. On the right side of the screen, there are system status indicators like battery level, signal strength, and the date/time (06-12-2021, 17:11).

To Create a Carts Dashboard:

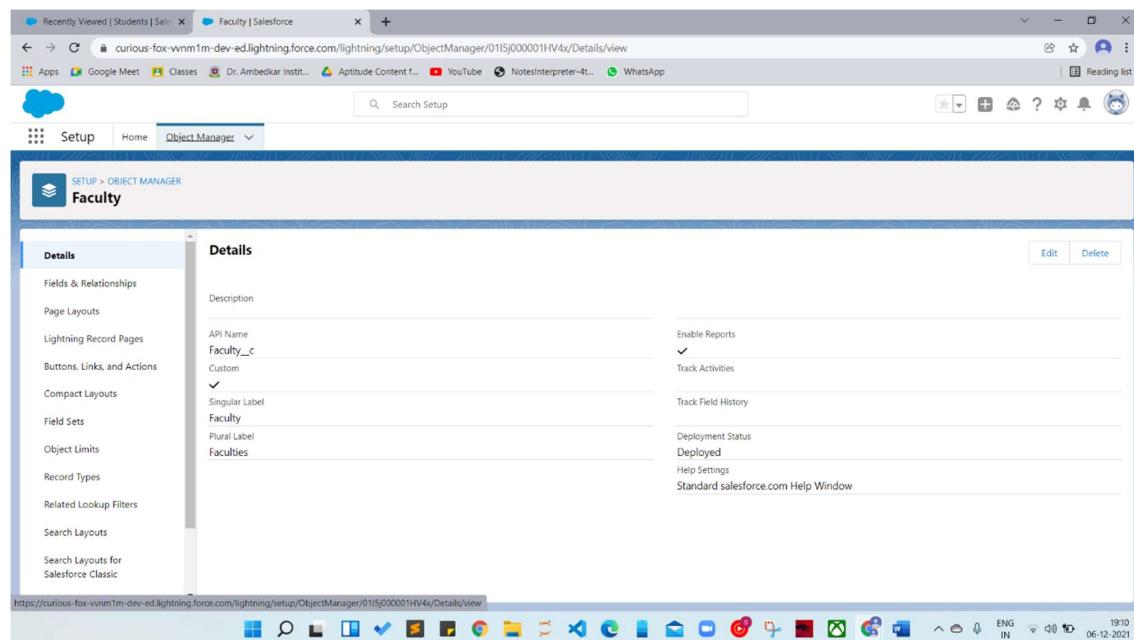
- Go to “Dashboard tab” and then click on “New Folder” and give it any Name.
- Click on “New Dashboard” and then name it as “Shopping cart” and select folder that you have created Click on Create.
- Click on the report that you have created and click on that and click select.
- Select any style to represent the data in dashboard.
- Add any filter(s), otherwise it is optional.
- Click on Save and Click Run



3. Create a web application to enter the faculty details like faculty ID, faculty name, and salary to a database and calculate the income tax to be paid by the faculty at the end of the financial year.

STEPS:

1. Launch your Salesforce Trailhead Playground by opening any module and Switch to Lightning Experience if you are currently in Salesforce Classic by clicking your picture in the right top corner and then click on “Switch to Lightning Experience”.
2. Then go to Setup gear icon and click “Setup”.
3. Click on “Object Manager” and click “Create> Custom Object” to create new Custom Object.



1. Name the object “Faculty”
2. Allow Reports and Allow Search.
3. Check the box in front of “Launch New Custom Tab Wizard after saving this custom object”
4. To create a Tab for the Object: Select any Tab Style for the object “Faculty”. Click Next, Next, leave the defaults and save.

5. To add fields to the Object: Go to “Fields & Relationships” option of Student object and Click “New”

6. Add the following fields one after the other:

- Field Label: ID (Length 10), Data Type: Text, provide an example ID as Help Text, make it as Required Field, don’t allow Duplicate Values, make it as Case Insensitive and Set this field as the unique record identifier from an external system
- Field Label: Salary, Data Type: Currency (Length 16, Decimal Places 2), Make it as Required Field

The screenshot shows the Salesforce Setup interface with the following details:

- Page Title:** Recently Viewed | Students | Sales > Faculty | Salesforce
- URL:** curious-fox-wmm1m-dev-ed.lightning.force.com/lightning/setup/ObjectManager/0115j000001HV4x/FieldsAndRelationships/view
- Section:** SETUP > OBJECT MANAGER
- Object:** Faculty
- Left Sidebar:** Details, Fields & Relationships (selected), Page Layouts, Lightning Record Pages, Buttons, Links, and Actions, Compact Layouts, Field Sets, Object Limits, Record Types, Related Lookup Filters, Search Layouts, Search Layouts for Salesforce Classic.
- Table:** Fields & Relationships (6 items, Sorted by Field Label)

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedById	Lookup(User)		
Faculty Name	Name	Text(50)		✓
ID	ID_c	Text(10) (External ID) (Unique Case Sensitive)		✓
Last Modified By	LastModifiedById	Lookup(User)		
Owner	OwnerId	Lookup(User.Group)		✓
Salary	Salary_c	Currency(16, 2)		
- Buttons:** Quick Find, New, Deleted Fields, Field Dependencies, Set History Tracking.
- Bottom:** Standard Salesforce navigation icons, Taskbar, and system status bar (ENG IN, 06-12-2021, 19:21).

To calculate Income Tax to be paid:

7. Go to “Buttons, Links and Actions” of Faculty Object and click “New Button or Link”

8. Name it as “Tax Calculation”

9. Select the radio button “Detail Page Link” as it is a website link.

10. Behaviour: Display in new window.

11. Content Source: URL.

12. Field Type: Faculty

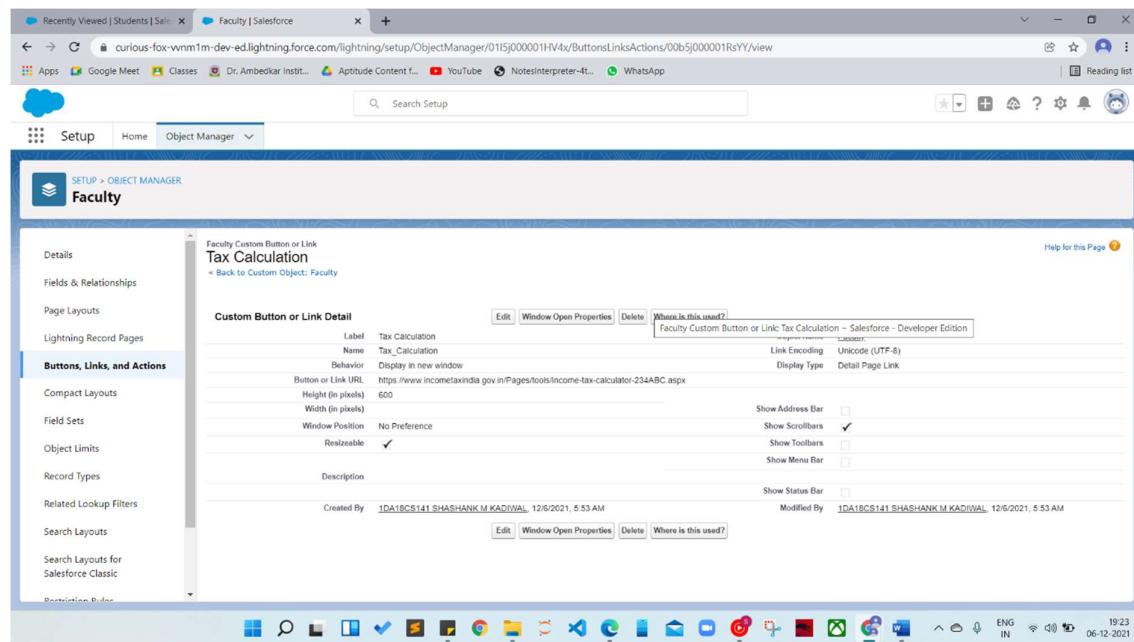
13. In the empty space provided, type

<https://www.incometaxindia.gov.in/Pages/tools/income-tax-calculator-234ABC.aspx>

It is a link which redirects to the income tax calculation website.

14. Link Encoding: Unicode (UTF-8).

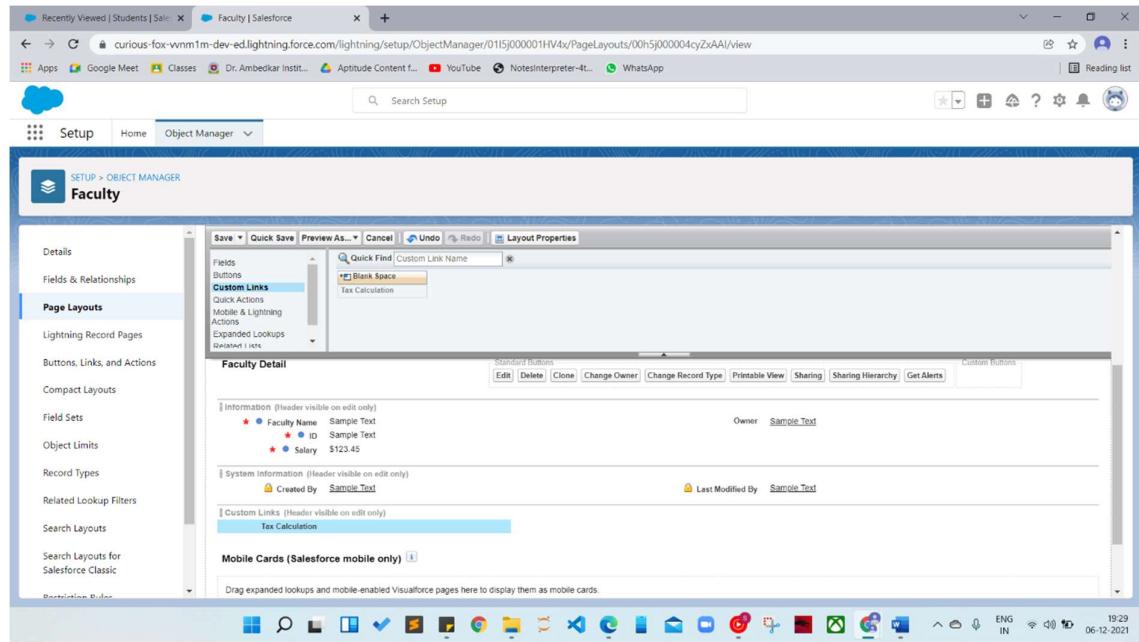
15. Click Save



16. Go to Page Layout, Click Faculty Layout.

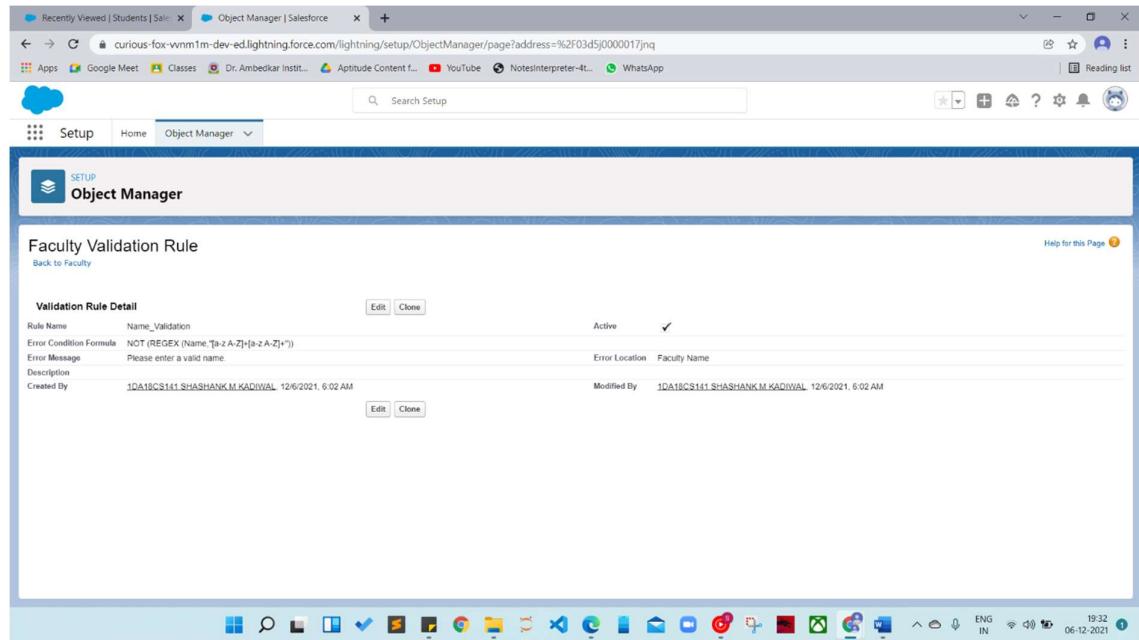
17. Click Custom Links, Drag and drop the “Tax Calculation” link in the Custom Link area.

18. Click Save.



To add a rule to the faculty name so that it should take only valid names:

- Go to Validation Rule of Faculty object and click “New”
- Name it as “Name Validation”.
- Error Condition Formula: NOT (REGEX (Name, ”[a-zA-Z]+ [a-zA-Z]+”)).
- Error Message: Please Enter a valid name.
- Error Location: Field – Faculty name.
- Click Save.



To create an application:

19. Go to “Setup” and type “App Manager” in Quick Find Box.
20. Click on “New Lightning App” to create a Lightning Application.
21. Name it as “Faculty Database”, give the description for your application.
22. Uploading Image and changing colours are optional, then click Next.
23. Navigation Style: Standard Navigation, click Next.
24. No need to add any Utility Bar, click Next.
25. Add the following Items: Faculties, Reports and Dashboards, click Next.
26. Assign it to System Administrator Profile by selecting System Administrator and pressing right arrow and then click Save & Finish.
27. Go to App Manager, select your application and select Faculties and click “New” to add some details to your application.
28. Click the entry you added, go to details. Make sure you will get an error message when you enter an invalid name and invalid id.

The screenshots illustrate the creation and viewing of a faculty record in a Salesforce application.

Screenshot 1: New Faculty Form

This screenshot shows the "New Faculty" form. The "Faculty Name" field contains "1Shashank M Kadiwal". The "ID" field contains "101". The "Salary" field contains "\$34,000.00". The "Owner" field shows "1DA18CS141 SHASHANK M KADIWAL". The "Save" button is visible at the bottom.

Screenshot 2: Faculty Record Details

This screenshot shows the details page for the faculty record "Shashank M Kadiwal". The "Details" tab is selected. The record shows the following information:

- Faculty Name: Shashank M Kadiwal
- ID: STAFF101
- Salary: \$34,000.00
- Created By: 1DA18CS141 SHASHANK M KADIWAL, 12/6/2021, 6:09 AM
- Last Modified By: 1DA18CS141 SHASHANK M KADIWAL, 12/6/2021, 6:09 AM
- A "Tax Calculation" link is present at the bottom left.

29. Press the “Tax Calculation” link to calculate income tax.
30. Click OK so that it will redirect you to the income tax calculator website.
31. Enter the required Details and press “Calculate”.

(As amended upto Finance Act, 2021)

INCOME AND TAX CALCULATOR

Click here to view relevant [Act & Rule](#).

Assessment Year	2022-23
Tax Payer	Individual
Whether opting for taxation under Section 115BAC?	Yes
Male / Female / Senior Citizen	Male
Residential Status	Resident
Income from Salary	34,000
Income From House Property <small>i</small>	0

Show Details

TAX INFORMATION AND SERVICES

TAX LAWS & RULES

INTERNATIONAL TAXATION

Reports and Dashboards:

To Create a Faculty Report:

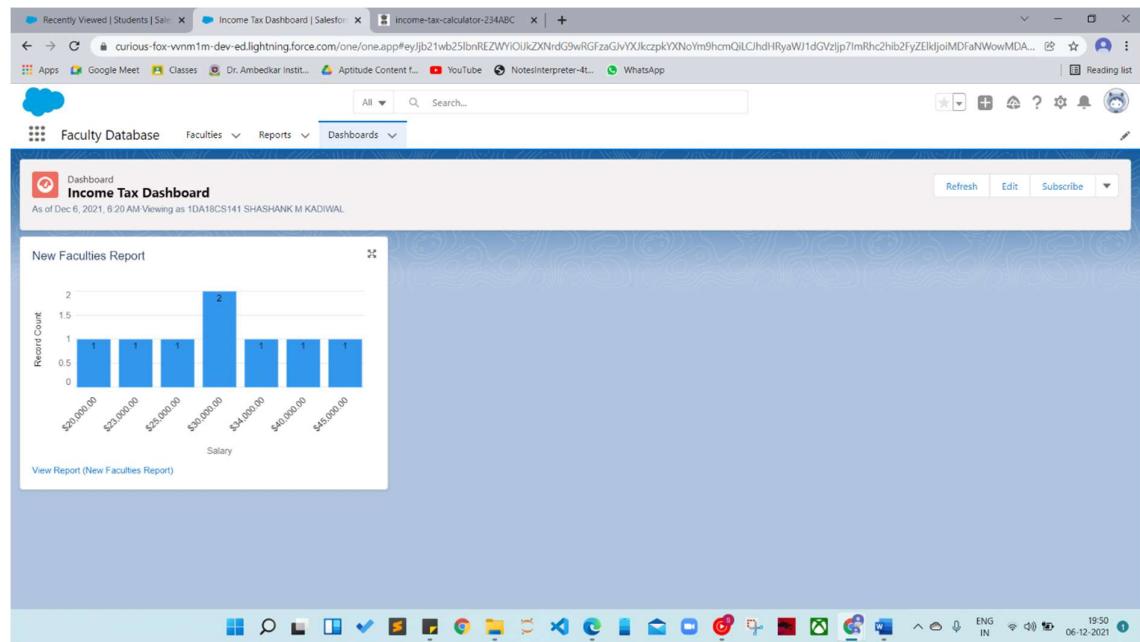
- Go to “Reports tab” Click on “New Folder” And give it any name and click on Save.
- Click on “New Report” and from search bar Search for “Faculty” and then select it then Click Continue.
- Add the required Columns to get the Complete Entered data.
- If you want the report to be grouped by any specific Fields then Search for the field in “Add groups” otherwise it is optional.
- Click on save and name the report as “New Faculties Report” and then select the folder which you have created.
- Click Save and then Click Run

The screenshot shows a Salesforce report interface titled "Report: Faculties New Faculties Report". The report displays a list of 8 faculty records. The columns are "Salary" (sorted ascending), "Faculty: Faculty Name", and "ID". The data is as follows:

Salary	Faculty: Faculty Name	ID
\$20,000.00 (1)	Rajev R	STAFF102
\$23,000.00 (1)	Akrili A	STAFF103
\$25,000.00 (1)	Suresh V	STAFF104
\$30,000.00 (2)	Uday Kumar V Dherendra	STAFF105 STAFF106
\$34,000.00 (1)	Shashank M Kadiwal	STAFF101
\$40,000.00 (1)	Sanjay	STAFF107
\$45,000.00 (1)	Saitaja	STAFF108

To Create a Faculty Dashboard:

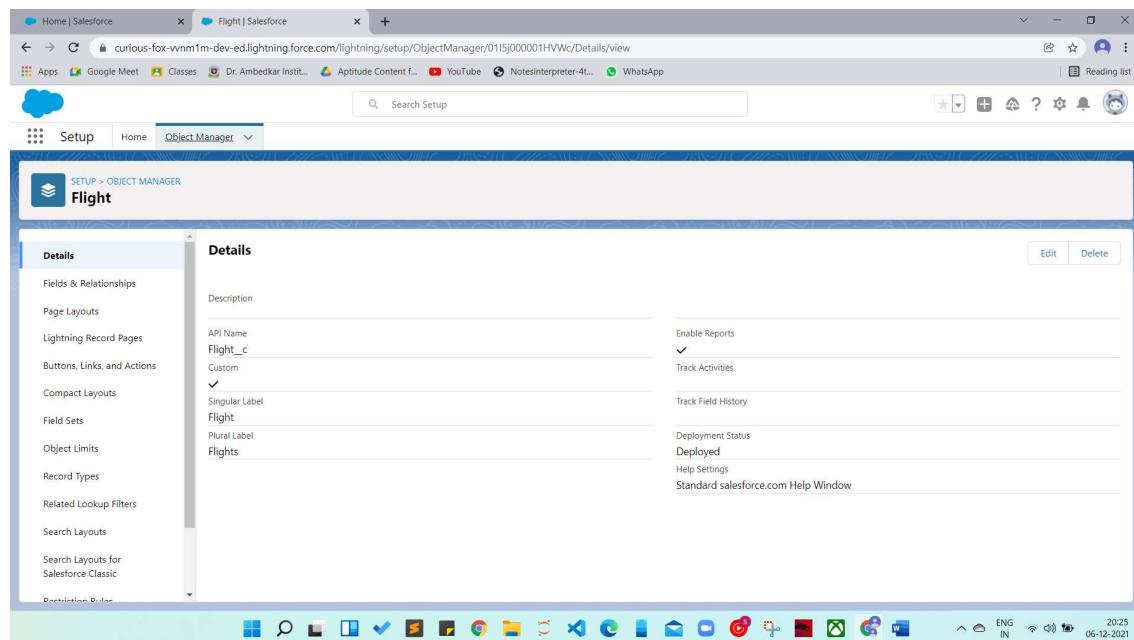
- Go to “Dashboard tab” and then click on “New Folder” and give it any Name.
- Click on “New Dashboard” and then name it as “Income Tax Dashboard” and select folder that you have created Click on Create.
- Click on the report that you have created and click on that and click select.
- Select any style to represent the data in dashboard.
- Add any filter(s), otherwise it is optional.
- Click on Save and Click Run



4.Create a web application to book a flight from a source to destination and store the status of flight, and departure timings on database.

STEPS:

1. Launch your Salesforce Trailhead Playground by opening any module and Switch to Lightning Experience if you are currently in Salesforce Classic by clicking your picture in the right top corner and then click on “Switch to Lightning Experience”
2. Then go to Setup gear icon and click “Setup”.
3. Click on “Object Manager” and click “Create > Custom Object” to create new Custom Object.
4. Name the object “Flight”.
5. Allow Reports and Allow Search.
6. Check the box in front of “Launch New Custom Tab Wizard after saving this custom object”.
7. To create a Tab for the Object: Select any Tab Style for the object “Flight”. Click Next, Next, leave the defaults and save.



8. To add fields to the Object: Go to “Fields & Relationships” option of Student object and Click “New”.

9. Add the following fields one after the other:

- Field Label: Source, Data Type: Text Area, make it as Required Field.
- Field Label: Destination, Data Type: Text Area, make it as Required Field.
- Field Label: Departure Timing, Data Type: Date/Time, make it as Required Field.

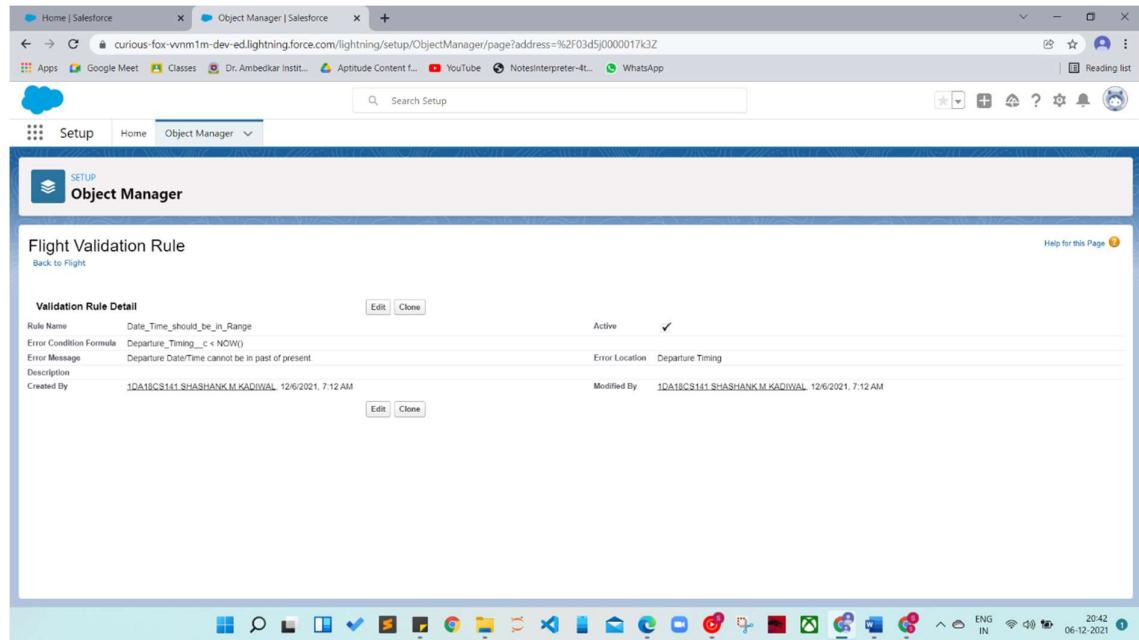
FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedById	Lookup(User)		
Departure Timing	Departure_Timing__c	Date/Time		
Destination	Destination__c	Text Area(255)		
Flight Name	Name	Text(80)		✓
Last Modified By	LastModifiedById	Lookup(User)		
Owner	OwnerId	Lookup(User.Group)		✓
Source	Source__c	Text Area(255)		

10. Validation Rules:

1. Date and time Validation:

To add a rule to the departure timing so that it is greater than today's date and the present time:

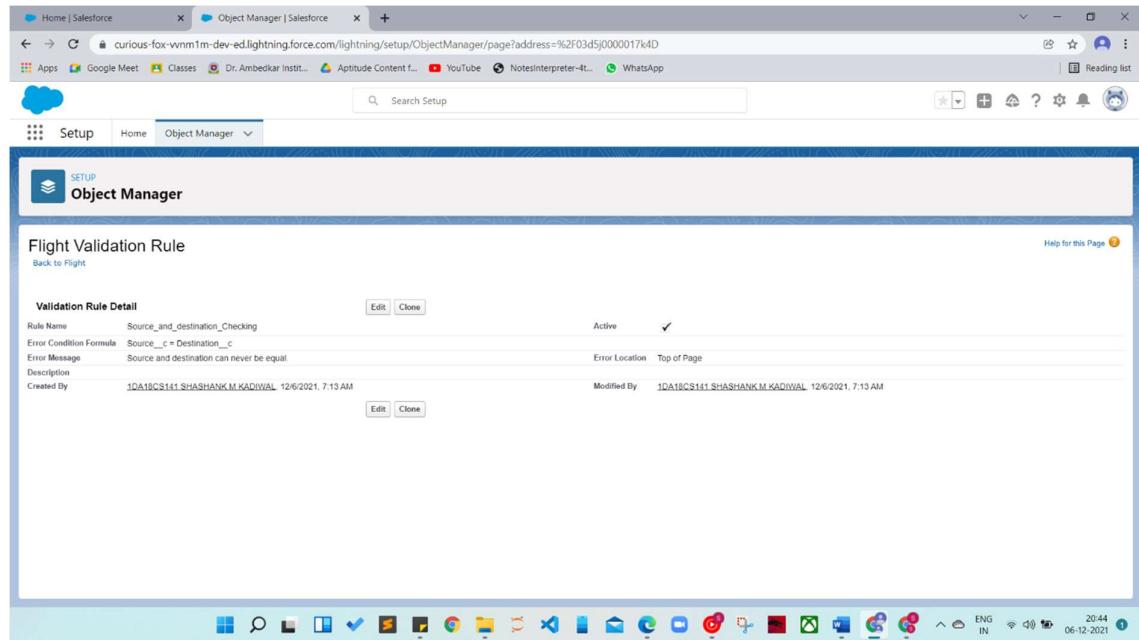
- Go to Validation Rule of Flight Object and click “New”
- Name it as “Date Time should be in Range”
- Error Condition Formula: `Departure_Timing__c < NOW()`
- Error Message: Departure Date/Time cannot be in past of present.
- Error Location: Field – Departure Timings.
- Click Save



2. Source and Destination Validation:

To add a rule to the Source and destination so that source and destination should not be equal:

- Go to Validation Rule of Flight Object and click “New”
- Name it as “Source and destination Checking”
- Error Condition Formula: `Source__c = Destination__c`
- Error Message: Source and destination can never be equal.
- Error Location: Top of the Page.
- Click Save.



Create one more object to provide status of the flight:

11. Name the Object “Status”
12. Allow Reports and Allow Search.
13. Check the box in front of “Launch New Custom Tab Wizard after saving this custom object”
14. Create a Tab for the Object.
15. To add fields to the Object: Go to “Fields & Relationships” option of Student object and Click “New”.
 - 16. Add the following fields one after the other:
 - Field Label: Flight Name, Data Type: Master-Detail Relationship, Related to: Flight.
 - Sharing Setting: Read-Only. Leave the defaults and save.
 - Master – Detail relationship is provided to enter status only to the existing flights.
 - Field Label: Flight Status, Data Type: Picklist, click radio button in front of Enter values, with each value separated by a new line.
 - Values are: Arrived, Cancelled, Delayed and Departed.

- Make it as Required Field and Restrict the values to the values in the picklist.

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedById	Lookup(User)		
Flight Name	Flight_Name__c	Master-Detail(Flight)		✓
Flight Status	Flight_Status__c	Picklist		✓
Last Modified By	LastModifiedById	Lookup(User)		✓
Status Name	Name	Text(80)		✓

17. To create an application:

18. Go to “Setup” and type “App Manager” in Quick Find Box.

19. Click on “New Lightning App” to create a Lightning Application.

20. Name it as “Flight Details”, give the description for your application.

21. Uploading Image and changing colours are optional, then click Next.

22. Navigation Style: Standard Navigation, click Next.

23. No need to add any Utility Bar, click Next.

24. Add the following Items: Flights, Statuses, Reports and Dashboards, click Next.

25. Assign it to System Administrator Profile by selecting System Administrator and pressing right arrow and then click Save & Finish.

26. Go to App Manager, select your application and select Flights and click “New” to add some details to your application.

The screenshot displays two overlapping Salesforce Lightning component dialogs on a desktop screen.

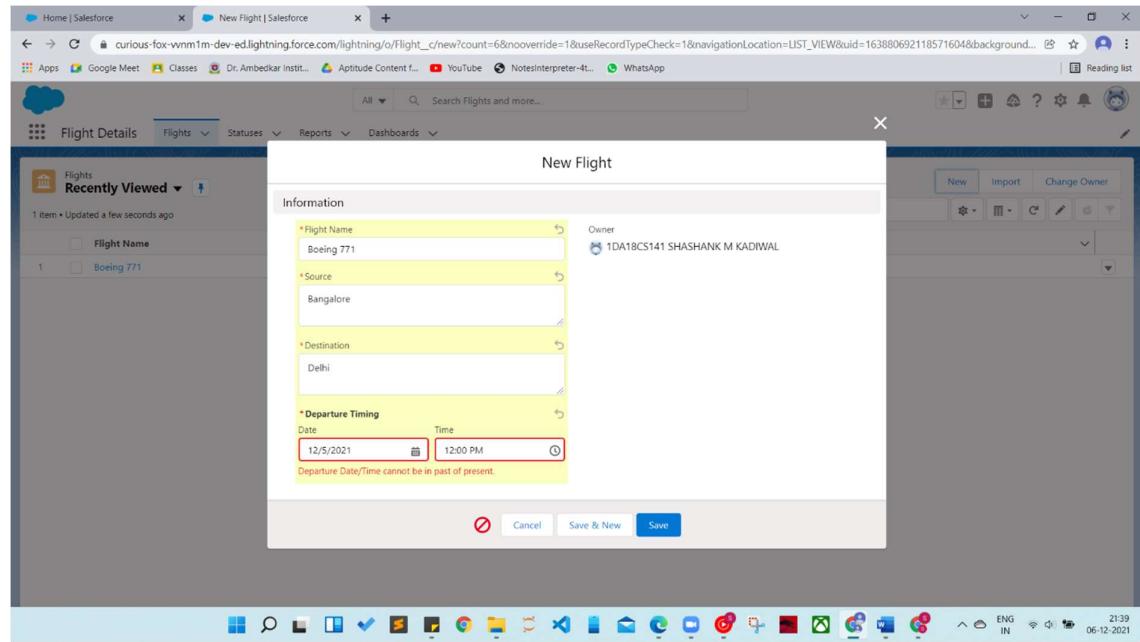
Edit Boeing 771 Dialog:

- Flight Name:** Boeing 771
- Source:** Bangalore
- Destination:** Mumbai
- Departure Timing:** Date: 12/6/2021, Time: 10:00 PM
- Created By:** 1DA18CS141 SHASHANK M KADIWAL, 12/6/2021, 8:04 AM
- Last Modified By:** 1DA18CS141 SHASHANK M KADIWAL, 12/6/2021, 8:04 AM

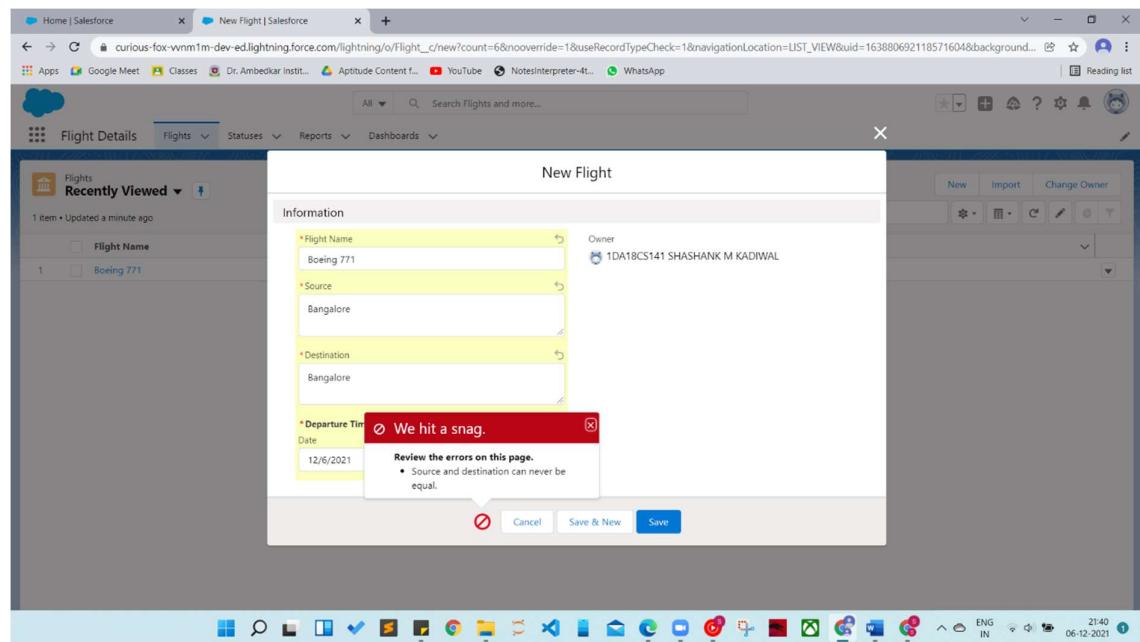
Edit Arrival Status Dialog:

- Status Name:** Arrival Status
- Flight Name:** Boeing 771
- Flight Status:** Arrived
- Created By:** 1DA18CS141 SHASHANK M KADIWAL, 12/6/2021, 8:05 AM
- Last Modified By:** 1DA18CS141 SHASHANK M KADIWAL, 12/6/2021, 8:05 AM

Make sure you will get an error message when you try to give the Departure Timing less than the current time and today's date



Make sure you will get an error message when you give source and destination station names as same.



Reports and Dashboards:

To Create a Flights Report:

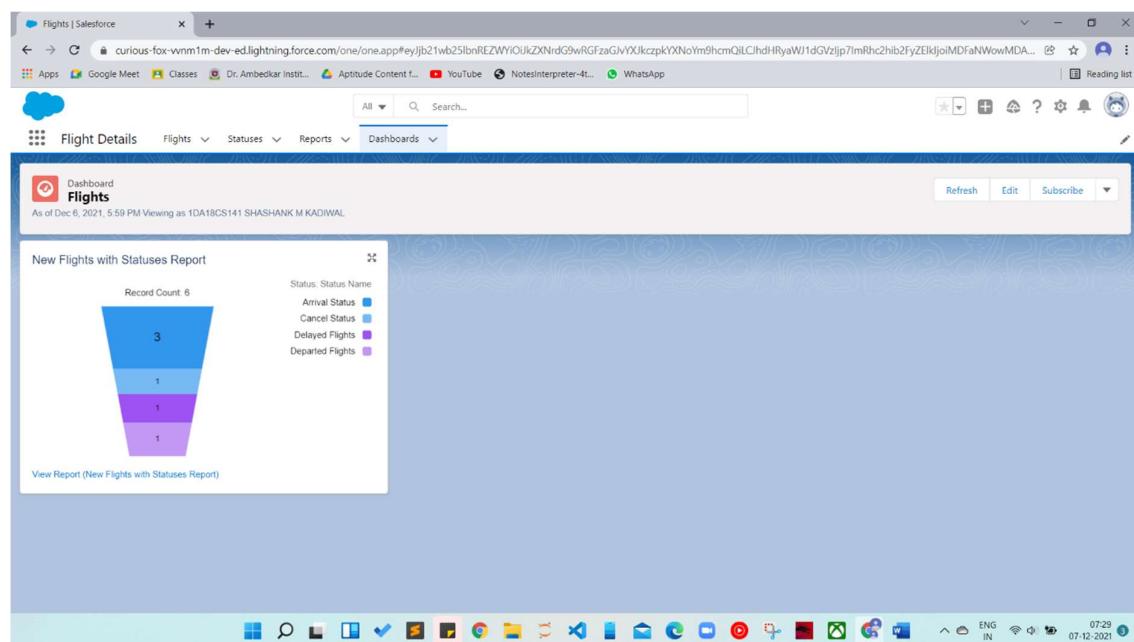
- Go to “Reports tab” Click on “New Folder” And give it any name and then click Save.
- Click on “New Report” and from search bar Search for “Flights” and then select it and Click Continue.
- Add the required Columns to get the Complete Entered data.
- If you want the report to be grouped by any specific Fields then Search for the field in “Add groups” otherwise it is optional.
- Click on save and name the report as “New Flights Report” and then select the folder which you have created.
- Click Save and then Click Run.

The screenshot shows a Salesforce Lightning Report titled "New Flights with Statuses Report". The report displays a table of flight status data with the following columns: Status, Status Name, Flight, and Flight Status. The data is grouped by Status Name, with subtotals for Arrival Status, Cancel Status, Delayed Flights, and Departed Flights. The total number of records is 6.

Status	Status Name	Flight	Flight Status
Arrival Status (3)	Lufthansa	Spicejet	Arrived
		Boeing 771	Arrived
Subtotal			
Cancel Status (1)	Air india		Cancelled
Subtotal			
Delayed Flights (1)	British Airways		Delayed
Subtotal			
Departed Flights (1)	Indigo		Departed
Subtotal			
Total (6)			

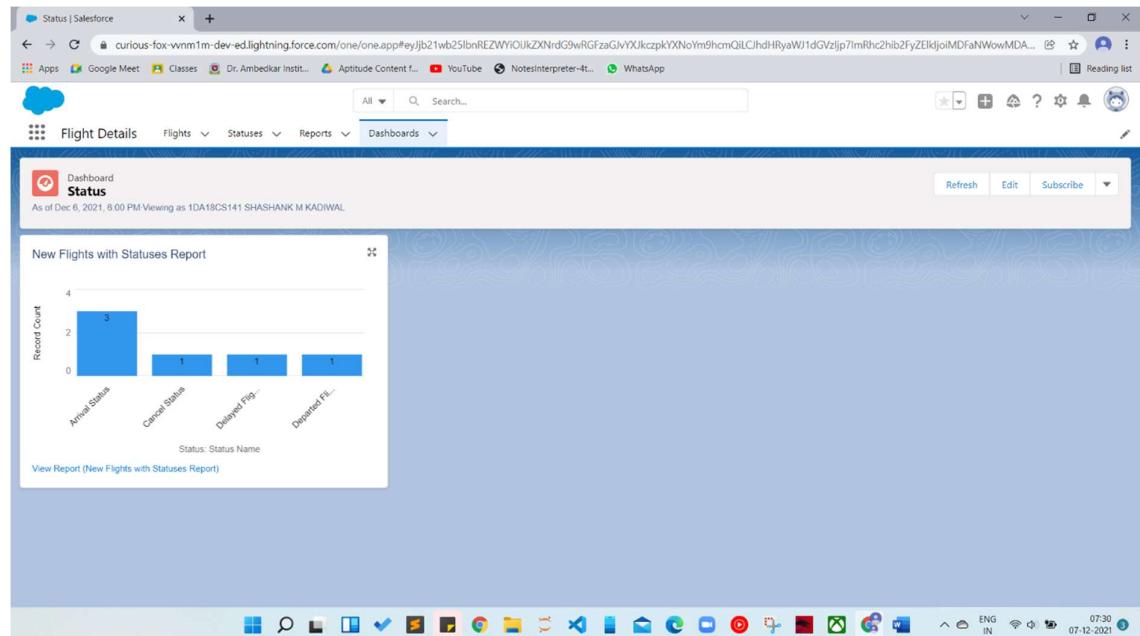
To Create a Status Report:

- Go to “Reports tab” Click on “New Folder” And give it any name and then click Save.
- Click on “New Report” and from search bar Search for “Flights with Status” and then select it and then click Continue.
- Add the required Columns to get the Complete Entered data.
- If you want the report to be grouped by any specific Fields then Search for the field in “Add groups” otherwise it is optional.
- Click on save and name the report as “New Flights with status Report” and then select the folder which you have created.
- Click Save and then Click Run



To Create a Status Dashboard:

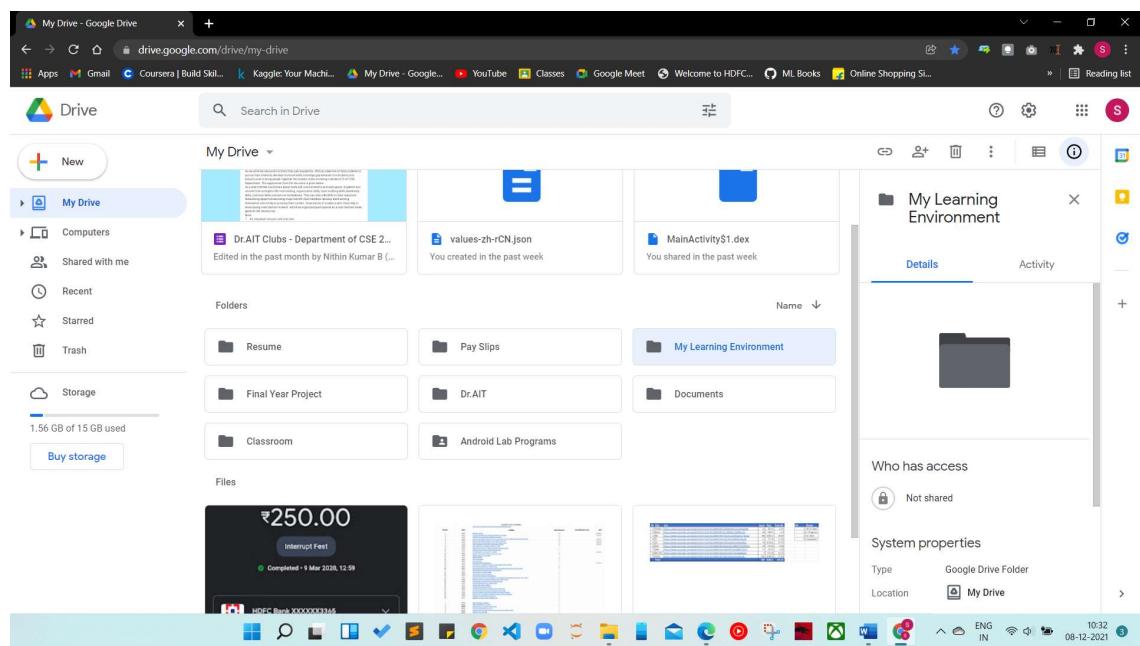
- Go to “Dashboard tab” and then click on “New Folder” and give it any Name.
- Click on “New Dashboard” and then name it as “Status” and select folder that you have created Click on Create.
- Click on the report that you have created and click on that and click select.
- Select any style to represent the data in dashboard.
- Add any filter(s), otherwise it is optional.
- Click on Save and Click Run



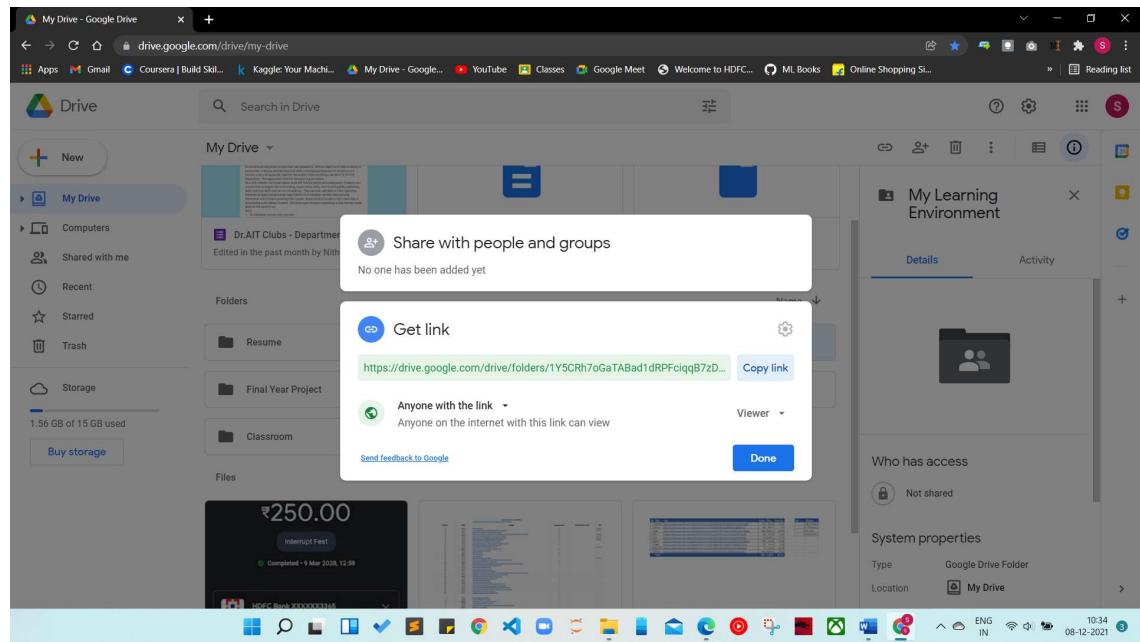
5.Create a collaborative learning environment for a particular learning topic using Google Apps. Google Drive, Google Docs and Google Slides must be used for hosting e-books, important articles and presentations.

STEPS:

1. Open <http://drive.google.com/> and Sign In with your google account.
2. Create a new folder named “My Learning Environment” by clicking “New” button on the top left corner.



3. Right click on the folder created and tap “Share” then click on “Advanced”.
4. Under the “Who has Access” section click on “Change” of the first option.
5. Now check on the “On- Public on the web” option & set the Access to “Can View Only” and Click Save. This will make your folder to be accessible by anyone on the internet to view its contents and download them



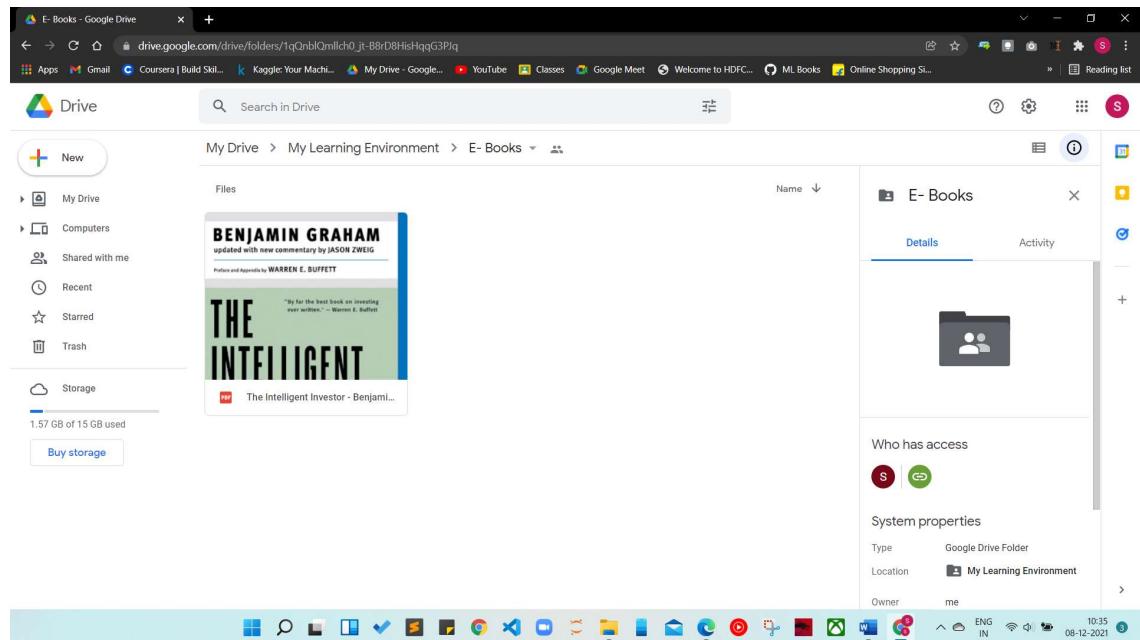
6. Copy the link and post it or share it to anyone you like.

Adding Learning Contents to your Environment:

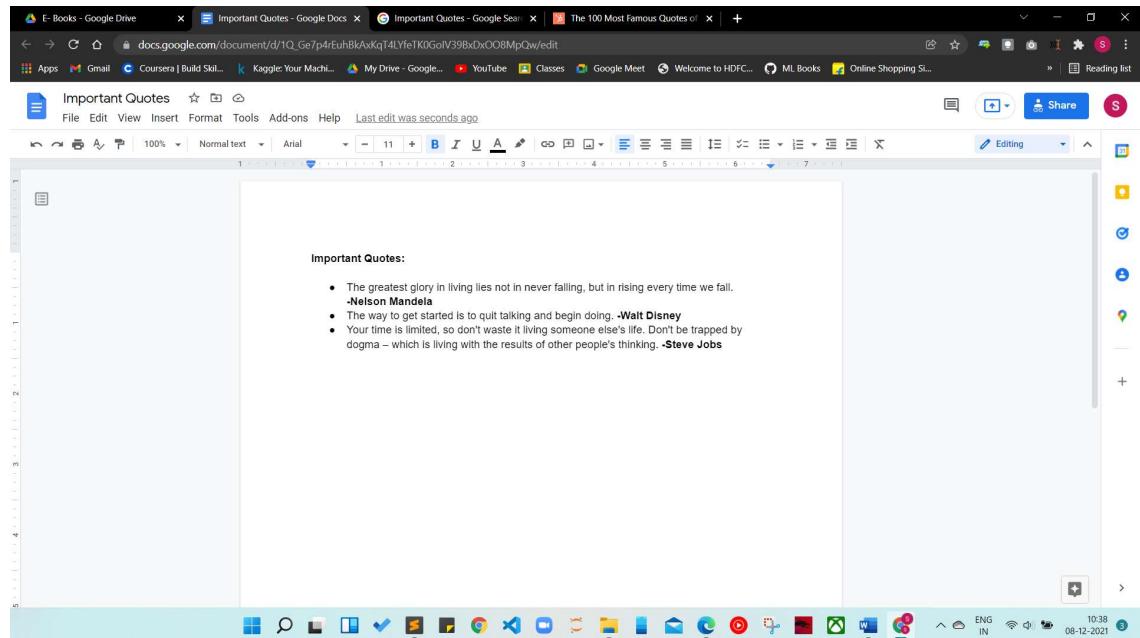
7. Double click on the folder you just created and click “New” button again.

8. Add these items:

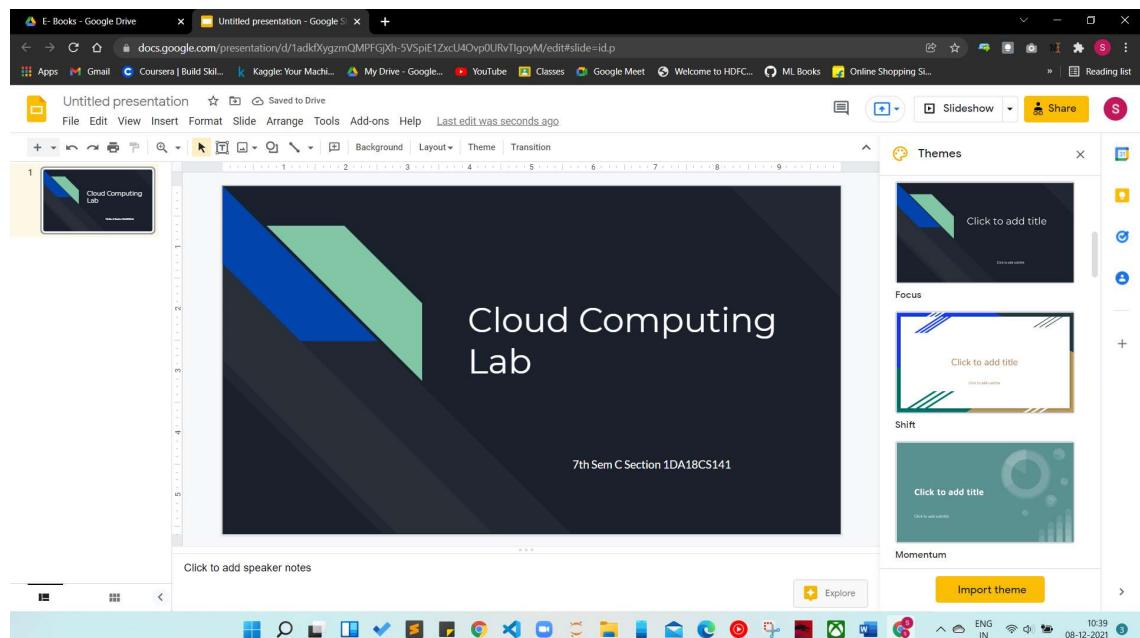
a. Folder: Named “E-Books” where you will open the folder and click “New” and “Upload a file” like a Book from your hard drive



b. Google Docs: Named “Important Quotes” where you will add some important links to the doc file. The file will keep saving hence you need to press save. click “Share” if you wish to share it



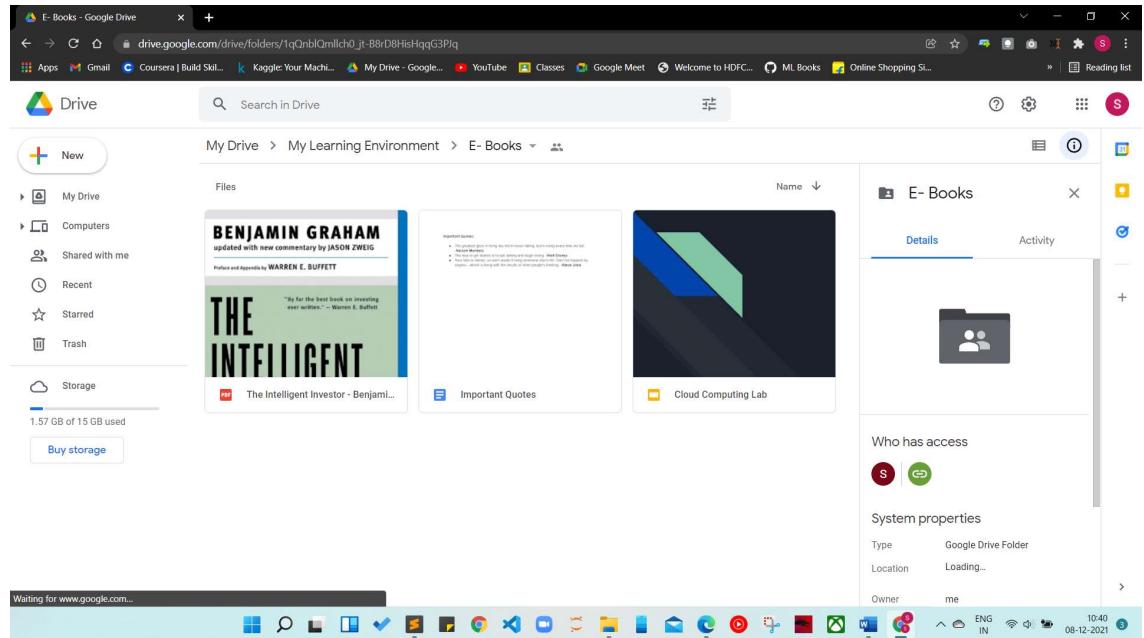
c. Google Slide: Named “Welcome Folks” make a few changes and add your content and choose your theme



d. You may also add many other items as necessary.

- At any point in future if you wish to share this environment right click on the folder and click share. Copy the link and share

https://drive.google.com/drive/folders/1qQnblQmllch0_jt-B8rD8HisHqqG3PJq?usp=sharing



6. Develop Department events registration app with an object containing event name, date/time, venue as parent relationship, another object containing student name, branch, event name, date/time, venue as child relationship.

STEPS:

1. Launch your Salesforce Trailhead Playground by opening any module and Switch to Lightning Experience if you are currently in Salesforce Classic by clicking your picture in the right top corner and then click on “Switch to Lightning Experience”.
2. Then go to Setup gear icon and click “Setup”.
3. Click on “Object Manager” and click “Create > Custom Object” to create new Custom Object.
4. Name the object “Event”.
5. Allow Reports and Allow Search.
6. Check the box in front of “Launch New Custom Tab Wizard after saving this custom object”.
7. To create a Tab for the Object:

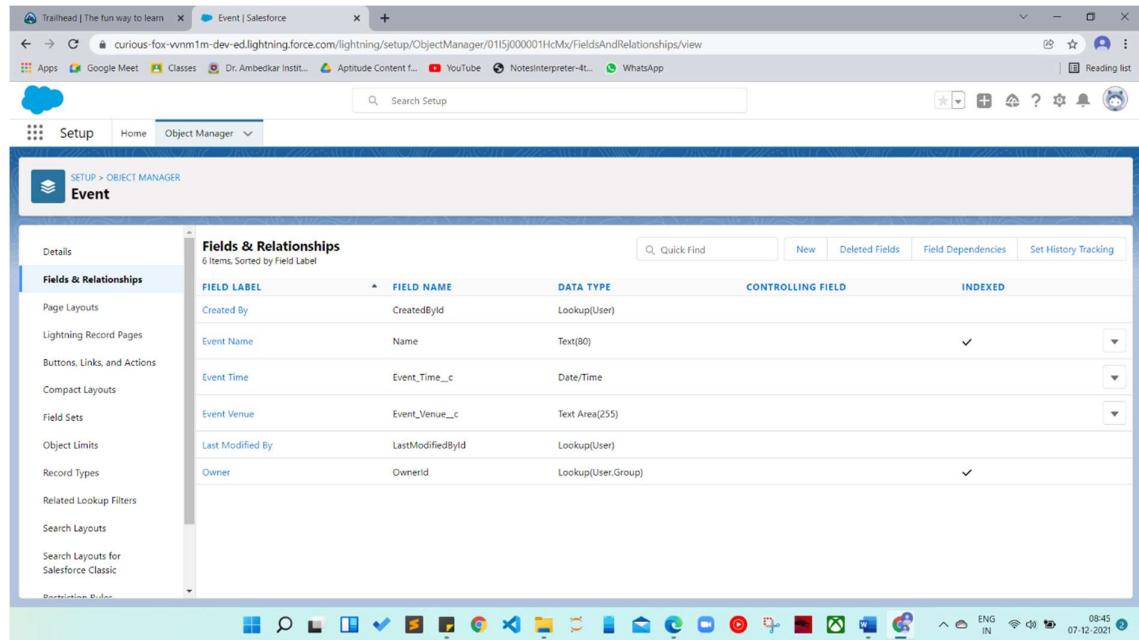
Select any Tab Style for the object “Event”. Click Next, Next, leave the defaults and save.

8. To add fields to the Object:

- Go to “Fields & Relationships” option of Student object and Click “New”.

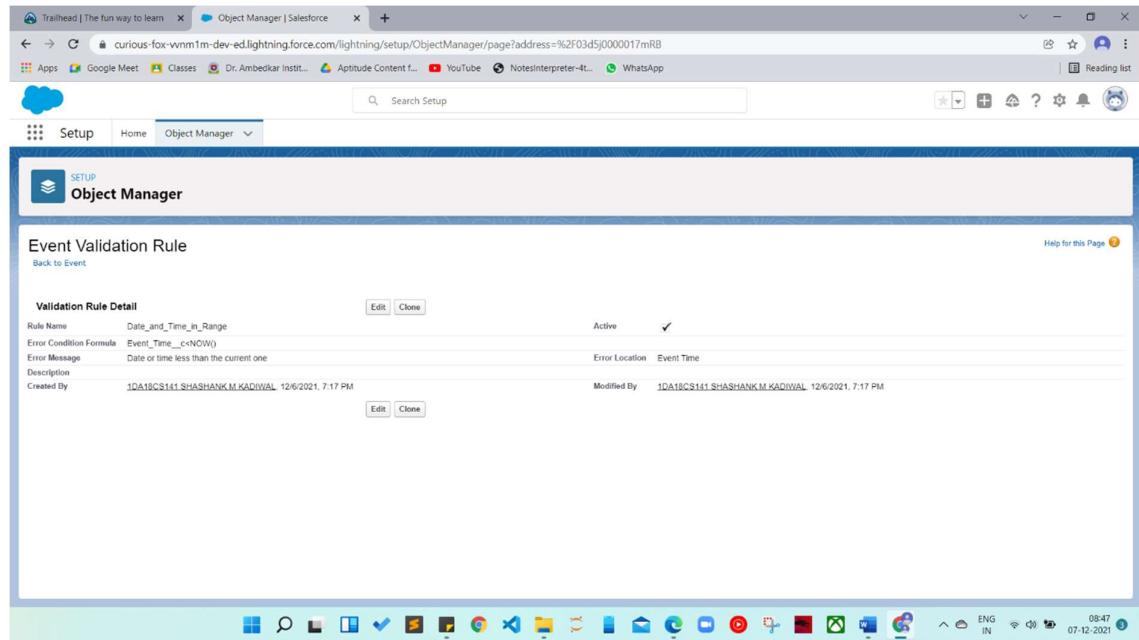
9. Add the following fields one after the other:

- Field Label: Event Time, Data Type: Date/Time, make it as Required Field.
- Field Label: Event Venue, Data Type: Text Area, make it as Required Field.



10. To add a rule to the Event Date/Time so that it is greater than today's date and the present time:

- Go to Validation Rule of Event Object and click “New”.
- Name it as “Date and Time in Range”.
- Error Condition Formula: `Event_Time__c < NOW ()`
- Error Message: Date or time less than the current one.
- Error Location: Field – Event Time.
- Click Save.



Create one more object to store student details:

11. Name the Object “Student1”.
12. Allow Reports and Allow Search.
13. Check the box in front of “Launch New Custom Tab Wizard after saving this custom object”.
14. Create a Tab for the Object.

15. To add fields to the Object:

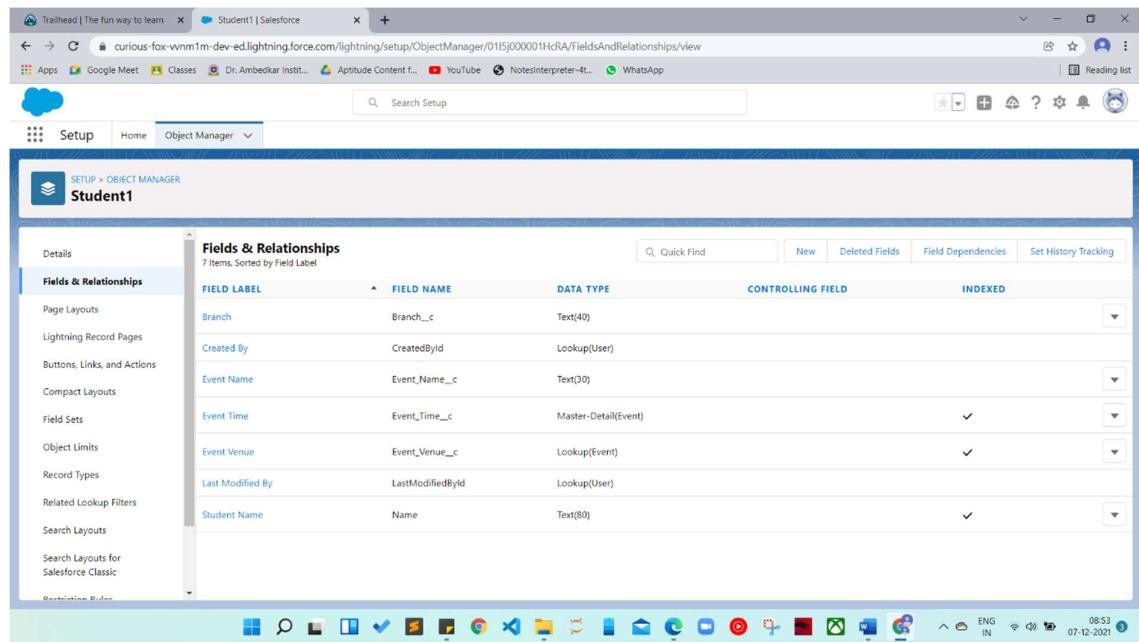
- Go to “Fields & Relationships” option of Student object and Click “New”.

16. Add the following fields one after the other:

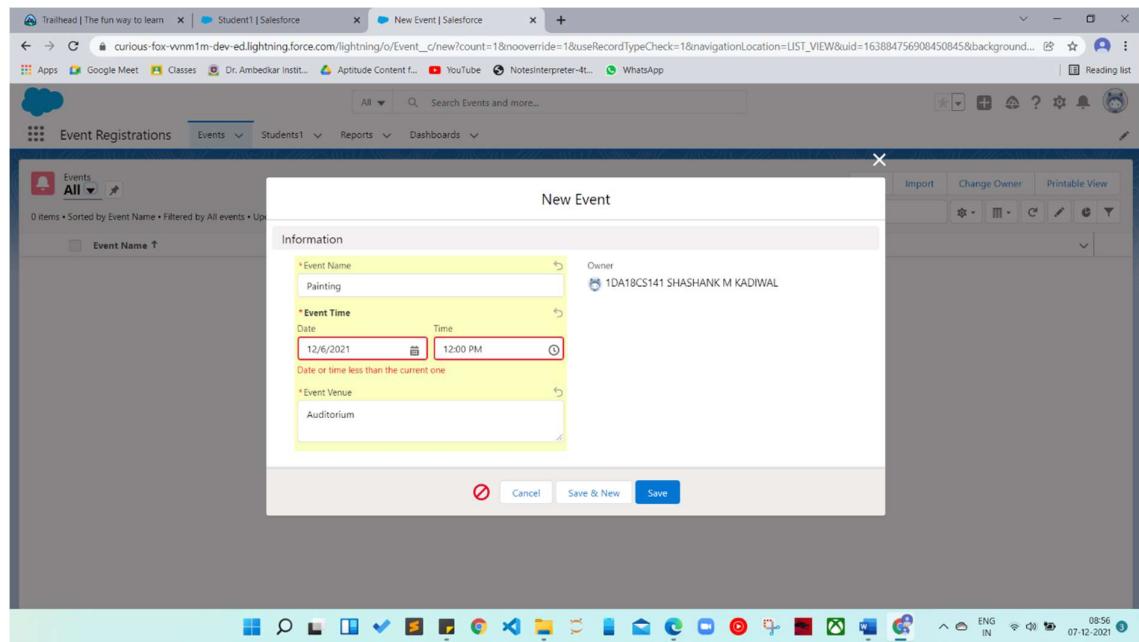
- Field Label: Event Name, Data Type: Text.
- Field Label: Event Time, Data Type: Master-Detail Relationship, Related to: Event.

Sharing Setting: Read-Only. Leave the defaults and save.

- Field Label: Event Venue, Data Type: Look up Relationship, Related to: Event.
- Field Label: Branch, Data Type: Text, Make it as a Required Field.



17. To create an application:
18. Go to “Setup” and type “App Manager” in Quick Find Box.
19. Click on “New Lightning App” to create a Lightning Application.
20. Name it as “Events’ Registrations”, give the description for your application.
21. Uploading Image and changing colours are optional, then click Next.
22. Navigation Style: Standard Navigation, click Next.
23. No need to add any Utility Bar, click Next.
24. Add the following Items: Events, Students, Reports and Dashboards, click Next.
25. Assign it to System Administrator Profile by selecting System Administrator and pressing right arrow and then click Save & Finish.
26. Go to App Manager, select your application and select Events and click “New” to add some details to your application.
27. Make sure you will get an error message when you try to give the Event Time less than the current time and today’s date and also you can’t select events which are not there in the list.



Reports and Dashboards:

To Create an Event Report:

- Go to “Reports tab” Click on “New Folder” And give it any name and Click Save.
- Click on “New Report” and from search bar Search for “Events” and then select it then Click Continue.
- Add the required Columns to get the Complete Entered data.
- If you want the report to be grouped by any specific Fields then Search for the field in “Add groups” otherwise it is optional.
- Click on save and name the report as “New Events Report” and then select the folder which you have created.
- Click Save and then Click Run

The screenshot shows a Salesforce report titled "Report: Events New Events Report". The report displays 9 total records. The data is presented in a table with three columns: "Event Venue", "Event: Event Name", and "Event Time".

Event Venue	Event: Event Name	Event Time
API Abdul Kalam Lab (1)	CodeVita	12/7/2021, 11:00 AM
Art Hall (1)	Drawing	12/8/2021, 11:00 AM
Auditions (1)	Dance Auditions	12/8/2021, 12:00 PM
Auditorium (3)	Painting	12/7/2021, 12:00 PM
	Fashion Show	12/7/2021, 3:00 PM
	Singing Auditions	12/7/2021, 2:00 PM
Room No : A213 (1)	Logic Quest	12/8/2021, 12:00 PM
Seminar Hall 214 (2)	Mock Interview	12/8/2021, 11:00 AM
	Tech Connect	12/8/2021, 2:00 PM

To Create a Students Report:

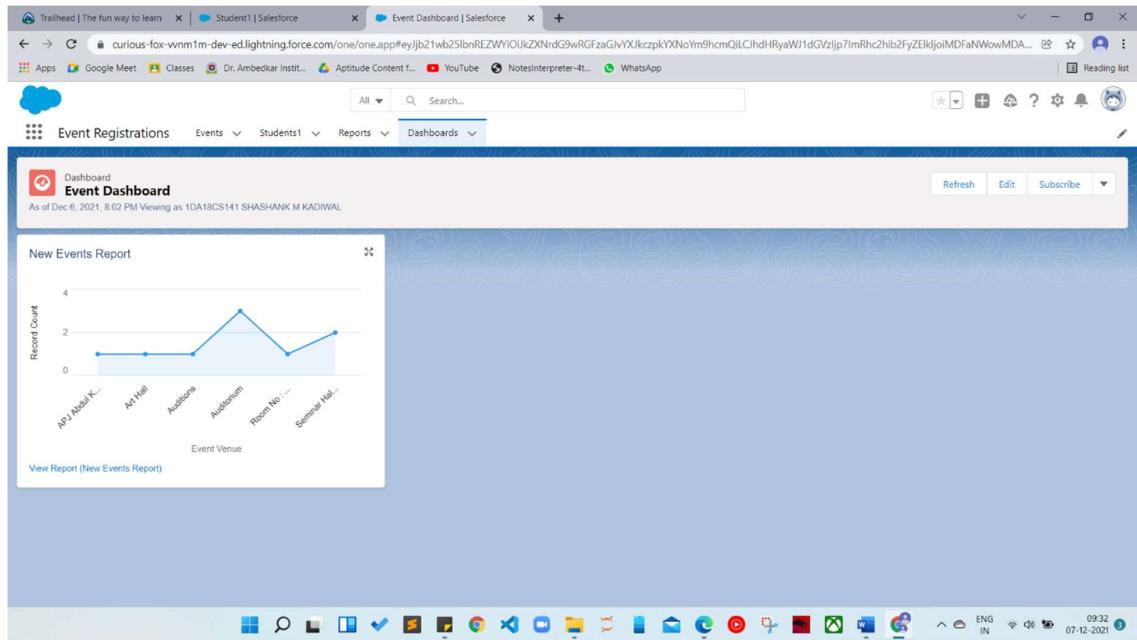
- Go to “Reports tab” Click on “New Folder” And give it any name and Click Save.
- Click on “New Report” and from search bar Search for “Students with Events” and then select it then Click Continue.
- Add the required Columns to get the Complete Entered data.
- If you want the report to be grouped by any specific Fields then Search for the field in “Add groups” otherwise it is optional.
- Click on save and name the report as “New Events with Students Report” and then select the folder which you have created

The screenshot shows a Salesforce Lightning interface with a report titled "New Events with Students1 Report". The report lists 6 total records. The columns are Branch, Student Name, Event Name, Event Time, and Event Venue. The data is categorized by branch:

Branch	Student Name	Event Name	Event Time	Event Venue
Computer Science (3)	Shashank M Kadiwal	CodeVita	12/7/2021, 11:00 AM	AP1 Abdul Kalam Lab
	Saranya Revanna	Logic Quest	12/8/2021, 12:00 PM	Room No : A213
	Salteja	Drawing	12/8/2021, 11:00 AM	Art Hall
Electronics (1)	Rajkumar	Singing Auditions	12/7/2021, 2:00 PM	Auditorium
Mechanical (1)	Rajendra	Fashion Show	12/7/2021, 3:00 PM	Auditorium
Medical Electronics (1)	Shrivatsa	Mock Interview	12/8/2021, 11:00 AM	Seminar Hall 214

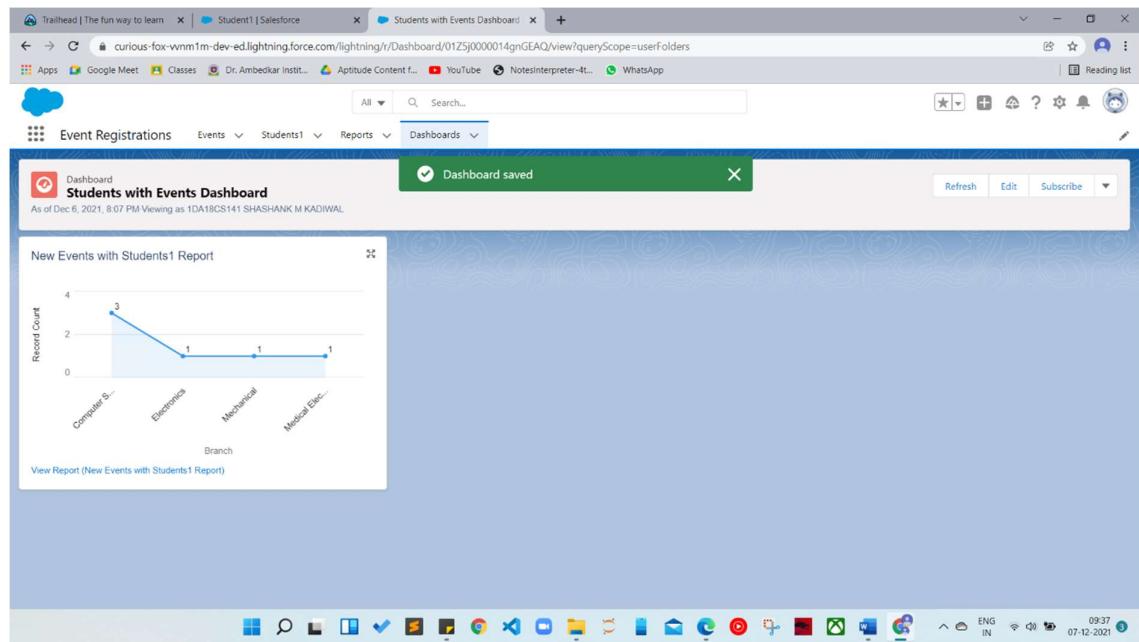
To Create an Event Dashboard:

- Go to “Dashboard tab” and then click on “New Folder” and give it any Name.
- Click on “New Dashboard” and then name it as “Event Dashboard” and select folder that you have created, Click on Create.
- Click on the report that you have created and click on that and click select.
- Select any style to represent the data in dashboard.
- Add any filter(s), otherwise it is optional.
- Click on Save and Click Run



To Create a Student1 Dashboard:

- Go to “Dashboard tab” and then click on “New Folder” and give it any Name.
- Click on “New Dashboard” and then name it as “Students with Events Dashboard” and select folder that you have created, Click on Create.
- Click on the report that you have created and click on that and click select.
- Select any style to represent the data in dashboard.
- Add any filter(s), otherwise it is optional.
- Click on Save and Click Run



7. Develop Blood Donation registration app with an object which records donors name, age and blood group as parent relationship and another object containing haemoglobin level, donated or not details.

STEPS:

1. Launch your Salesforce Trailhead Playground by opening any module and Switch to Lightning Experience if you are currently in Salesforce Classic by clicking your picture in the right top corner and then click on “Switch to Lightning Experience”
2. Then go to Setup gear icon and click “Setup”.
3. Click on “Object Manager” and click “Create > Custom Object” to create new Custom Object.
4. Name the object “Donor”
5. Allow Reports and Allow Search.
6. Check the box in front of “Launch New Custom Tab Wizard after saving this custom object”.
7. To create a Tab for the Object: Select any Tab Style for the object “Donor”. Click Next, Next, leave the defaults and save.
8. To add fields to the Object:
Go to “Fields & Relationships” option of Student object and Click “New”.
9. Add the following fields one after the other:
 - Field Label: Age, Data Type: Number (3,0).
 - Field Label: Blood Group, Data Type: Picklist, click radio button in front of Enter values, with each value separated by a new line.
 - Values are: A+ve, B+ve, A-ve, B-ve, O+ve, O-ve, AB+ve, AB-ve.
 - Make it as Required Field and Restrict the values to the values in the Picklist.
 - Field Label: Gender, Data Type: Picklist, click radio button in front of Enter values, with each value separated by a new line.

- Values are: Male, Female, Others.
- Field Label: Weight, Data Type: Number (3,2).

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Age	Age__c	Number(3, 0)		
Blood Group	Blood_Group__c	Picklist		
Created By	CreatedById	Lookup(User)		
Donor Name	Name	Text(80)		
Gender	Gender__c	Picklist		
Last Modified By	LastModifiedById	Lookup(User)		
Owner	OwnerId	Lookup(User,Group)		
Weight	Weight__c	Number(3, 2)		

Create one more object to store Collection details:

10. Name the Object “Record”.
 11. Allow Reports and Allow Search.
 12. Check the box in front of “Launch New Custom Tab Wizard after saving this custom object”
 13. Create a Tab for the Object.
 14. To add fields to the Object:
 - Go to “Fields & Relationships” option of Student object and Click “New”.
 15. Add the following fields one after the other:
 - Field Label: Haemoglobin level, Data Type: Number (2,2).
 - Field Label: Donor Name, Data Type: Look up Relationship, Related to: Donor.
 - Field Label: Blood Group, Data Type: Look up Relationship, Related to: Donor.
 - Field Label: Age, Data Type: Master-Detail Relationship, Related to: Donor.
- Sharing Setting: Read-Only. Leave the defaults and save.

- Field Label: Status, Data Type: Picklist, Values: Donated, Not Donated, Make it as a Required Field

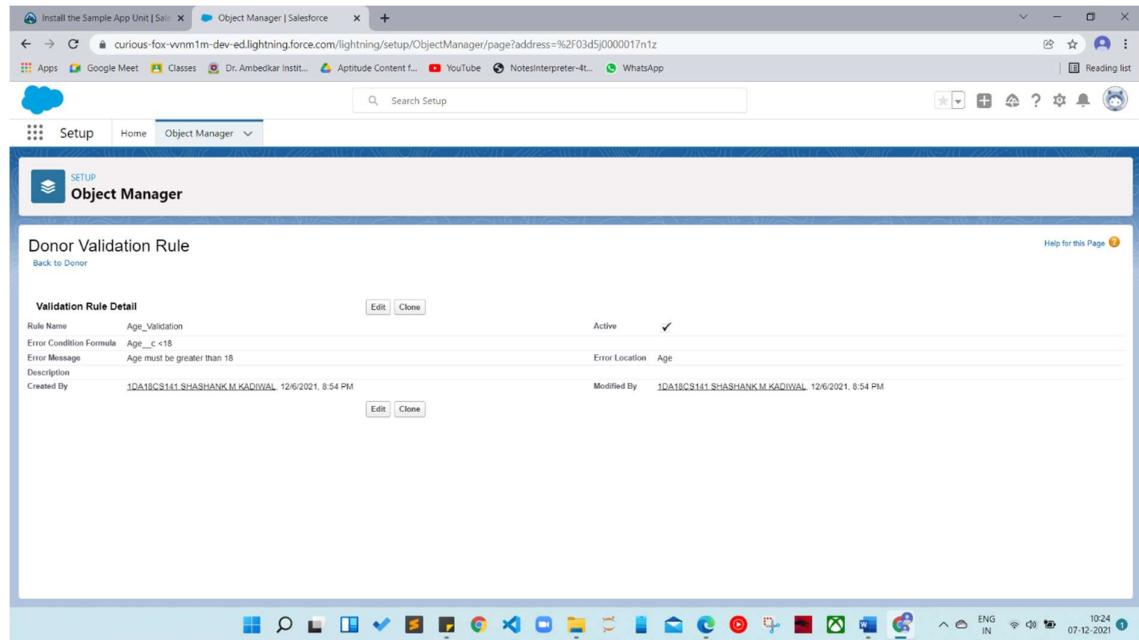
The screenshot shows the Salesforce Object Manager interface. The left sidebar has 'Fields & Relationships' selected under 'Record'. The main area displays a table titled 'Fields & Relationships' with the following data:

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Age	Age_c	Master-Detail(Donor)		✓
Blood Group	Blood_Group__c	Lookup(Donor)		✓
Created By	CreatedById	Lookup(User)		
Donor Name	Donor_Name__c	Lookup(Donor)		✓
Haemoglobin level	Haemoglobin_level__c	Number(2, 2)		
Last Modified By	LastModifiedById	Lookup(User)		
Re Name	Name	Text(80)		✓
Status	Status__c	Picklist		

Validation Rules:

1. Age Validation: To add a rule to the Donor age so that it is greater than 18 years:

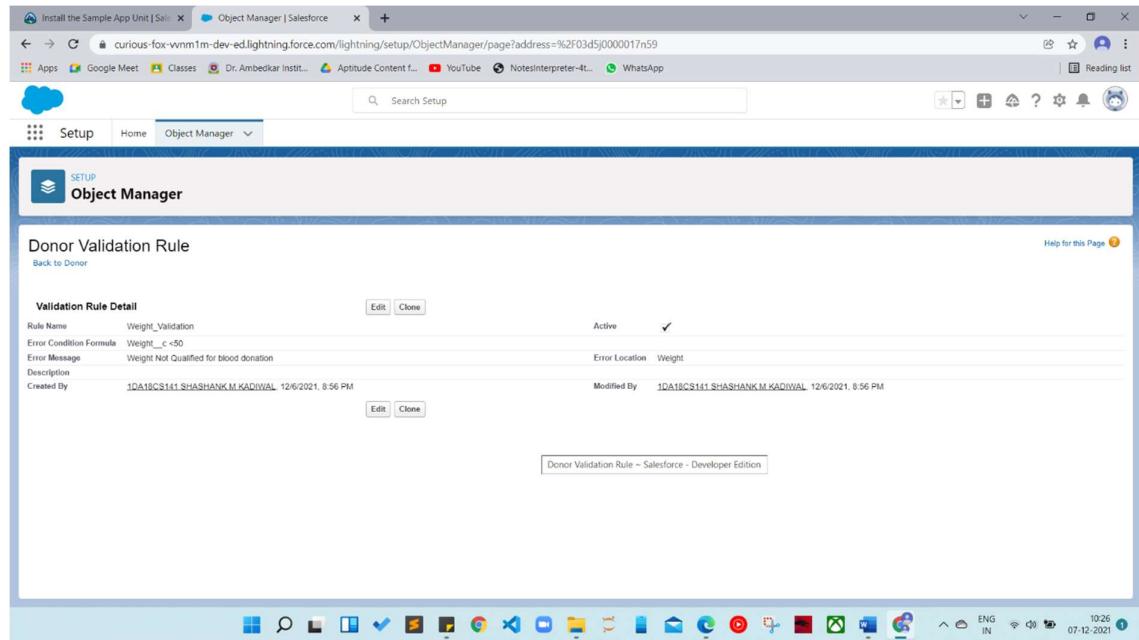
- Go to Validation Rule of Donor Object and click “New”
- Name it as “Age Validation”.
- Error Condition Formula: `age__c < 18.`
- Error Message: Age must be greater than 18.
- Error Location: Field – Age.
- Click Save



2. Weight Validation:

To add a rule to the Donor Weight so that it Should be greater than 50:

- Go to Validation Rule of Donor Object and click “New”
- Name it as “Weight Validation”.
- Error Condition Formula: weight__c < 50.
- Error Message: Eat more and gain your weight to 50 kgs.
- Error Location: Field –weight.
- Click Save



16. To create an application:
17. Go to “Setup” and type “App Manager” in Quick Find Box.
18. Click on “New Lightning App” to create a Lightning Application.
19. Name it as “Blood Donation”, give the description for your application.
20. Uploading Image and changing colours are optional, then click Next.
21. Navigation Style: Standard Navigation, click Next.
22. No need to add any Utility Bar, click Next.
23. Add the following Items: Donors, Records, Reports and Dashboards, click Next.
24. Assign it to System Administrator Profile by selecting System Administrator and pressing right arrow and then click Save & Finish.
25. Go to App Manager, select your application and select Donors and click “New” to add some details to your application.
26. Make sure you can’t donate when your age is less than 18 years and when your weight is not 50 kgs.

The screenshot shows the 'New Donor' form in the Salesforce Lightning interface. The form includes fields for Donor Name, Age, Blood Group, Gender, and Weight. The 'Age' field contains '15' and has a validation error message 'Age must be greater than 18'. The 'Weight' field contains '48.00' and has a validation error message 'Weight Not Qualified for blood donation'. The 'Save' button is at the bottom right.

The screenshot shows a confirmation message: 'Donor [Shashank M Kadiwal] was created.' This message is displayed above the donor's details, which include fields like Donor Name, Age, Blood Group, Gender, and Weight, along with the owner information and last modified details.

Reports and Dashboards:

To Create a Donor Report:

- Go to “Reports tab” Click on “New Folder” And give it any name then click on Save.
- Click on “New Report” and from search bar Search for “Donors” and then select it and then click on Continue.

- Add the required Columns to get the Complete Entered data
- If you want the report to be grouped by any specific Fields then Search for the field in “Add groups” otherwise it is optional.
- Click on save and name the report as “New Donor Report” and then select the folder which you have created.
- Click Save and then Click Run.

The screenshot shows a web browser window with a Salesforce Lightning report titled "New Donors with Records Report". The report displays a table of donor records. The table has columns for Blood Group, Donor Name, Age, Weight, and Gender. The data shows 7 total records. The browser window also shows other tabs and a toolbar at the top.

Blood Group	Donor Name	Age	Weight	Gender
A+ve (2)	Shashank M Kadiwal	19	54.00	Male
	Pramod P	18	59.00	Male
B+ve (1)	Deepika P	35	70.00	Female
A-ve (1)	Manish M	20	56.00	Male
O-ve (1)	Mahadev R	67	68.00	Male
AB+ve (2)	Nihal	25	50.00	Male
	Pushpa K	58	60.00	Female

To Create a Records Report:

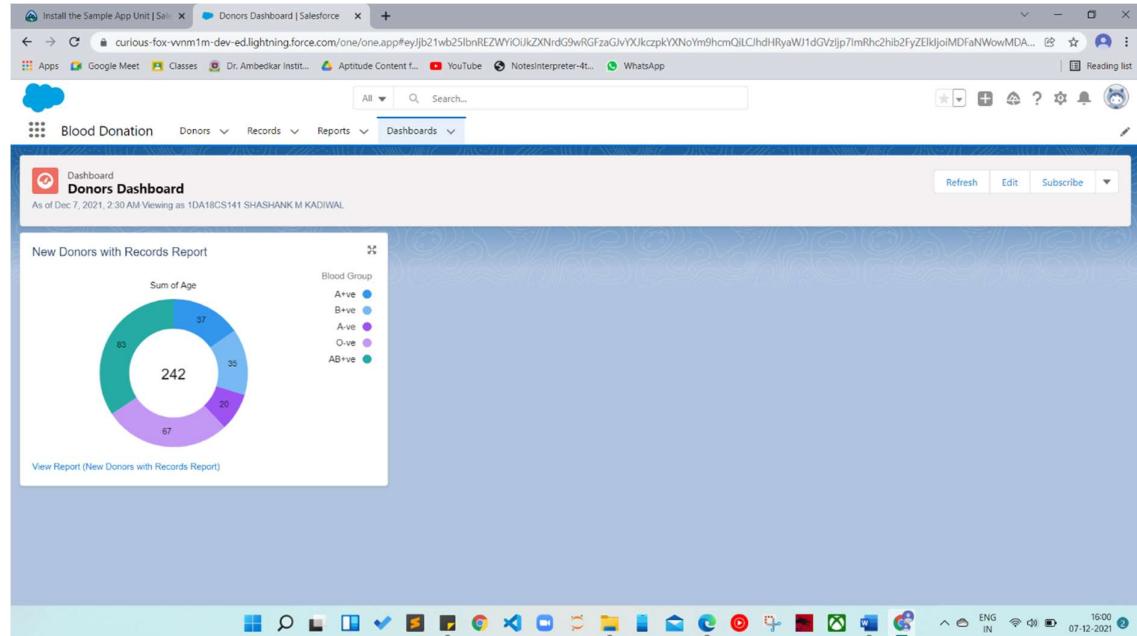
- Go to “Reports tab” Click on “New Folder” And give it any name then click on Save.
- Click on “New Report” and from search bar Search for “Donors with records” and then select it and then click on Continue.
- Add the required Columns to get the Complete Entered data.
- If you want the report to be grouped by any specific Fields then Search for the field in “Add groups” otherwise it is optional.
- Click on save and name the report as “New Donors with record Report” and then select the folder which you have created.
- Click Save and then Click Run.

Status	Donor Name	Record: Re Name	Blood Group	Haemoglobin level
Donated (3)	Pramod P	Pramod P	A+ve	17.80
	Nihal	Nihal	AB+ve	14.50
	Pushpa K	Pushpa K	AB+ve	16.30
Not Donated (2)	Deepika P	Deepika P	B+ve	15.00
	Shashank M Kadiwal	Shashank M Kadiwal	A+ve	16.00

To Create a Donor Dashboard:

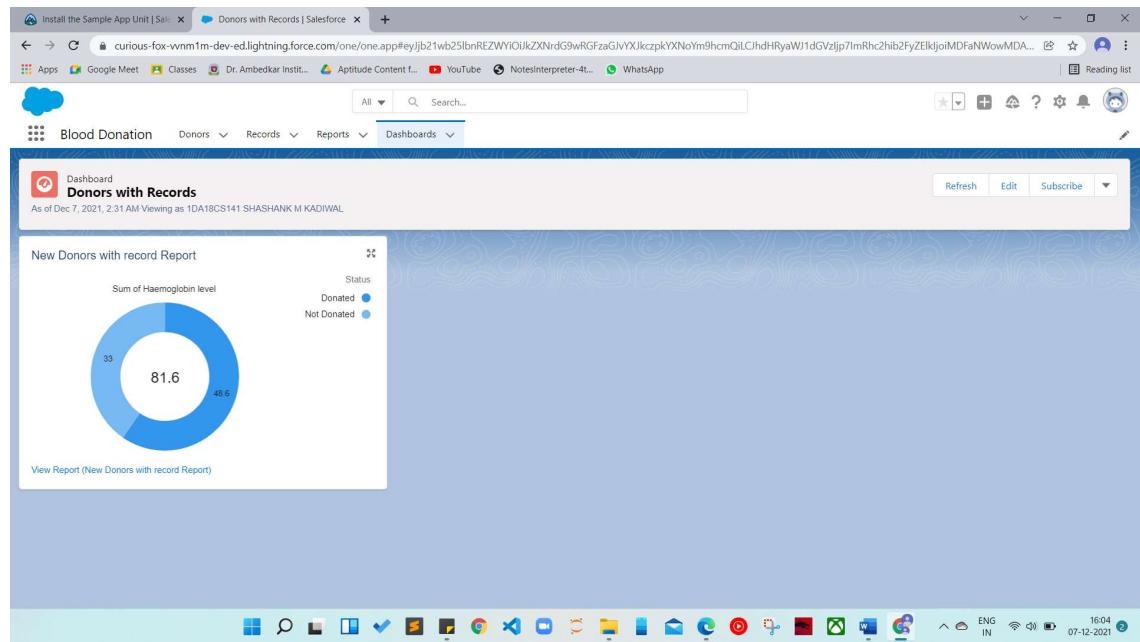
- Go to “Dashboard tab” and then click on “New Folder” and give it any Name.
- Click on “New Dashboard” and then name it as “Donors Dashboard” and select folder that you have created Click on Create.
- Click on the report that you have created and click on that and click select.

- Select any style to represent the data in dashboard.
- Add any filter(s), otherwise it is optional.
- Click on Save and Click Run



To Create a Records Dashboard:

- Go to “Dashboard tab” and then click on “New Folder” and give it any Name.
- Click on “New Dashboard” and then name it as “Donors with record Dashboard” and select folder that you have created Click on Create.
- Click on the report that you have created and click on that and click select.
- Select any style to represent the data in dashboard.
- Add any filter(s), otherwise it is optional.
- Click on Save and Click Run



8. Develop Attendance maintenance app with an object to record student details and attendance and a provide a link to college websites' results webpage.

1. Launch your Salesforce Trailhead Playground by opening any module and Switch to Lightning Experience if you are currently in Salesforce Classic by clicking your picture in the right top corner and then click on “Switch to Lightning Experience”
2. Then go to Setup gear icon and click “Setup”.
3. Click on “Object Manager” and click “Create > Custom Object” to create new Custom Object.
4. Name the object “Student”.
5. Allow Reports and Allow Search.
6. Check the box in front of “Launch New Custom Tab Wizard after saving this custom object”.
7. To create a Tab for the Object: Select any Tab Style for the object “Student”. Click Next, Next, leave the defaults and save.
8. To add fields to the Object:
Go to “Fields & Relationships” option of Student object and Click “New”.
9. Add the following fields one after the other:
 - Field Label: USN (Length 10), Data Type: Text, provide an example USN as Help Text, make it as Required Field and Don’t allow Duplicate Values and make it as Case Insensitive.
 - Field Label: Technical Branches, Data Type: Picklist, click radio button in front of Enter values, with each value separated by a new line.
 - Values are: CSE, ISE, TCE, EEE, EC etc.
 - Make it as Required Field and Restrict the values to the values in the picklist.
 - Field Label: Total Attendance %, Data Type: Percent (3,2).

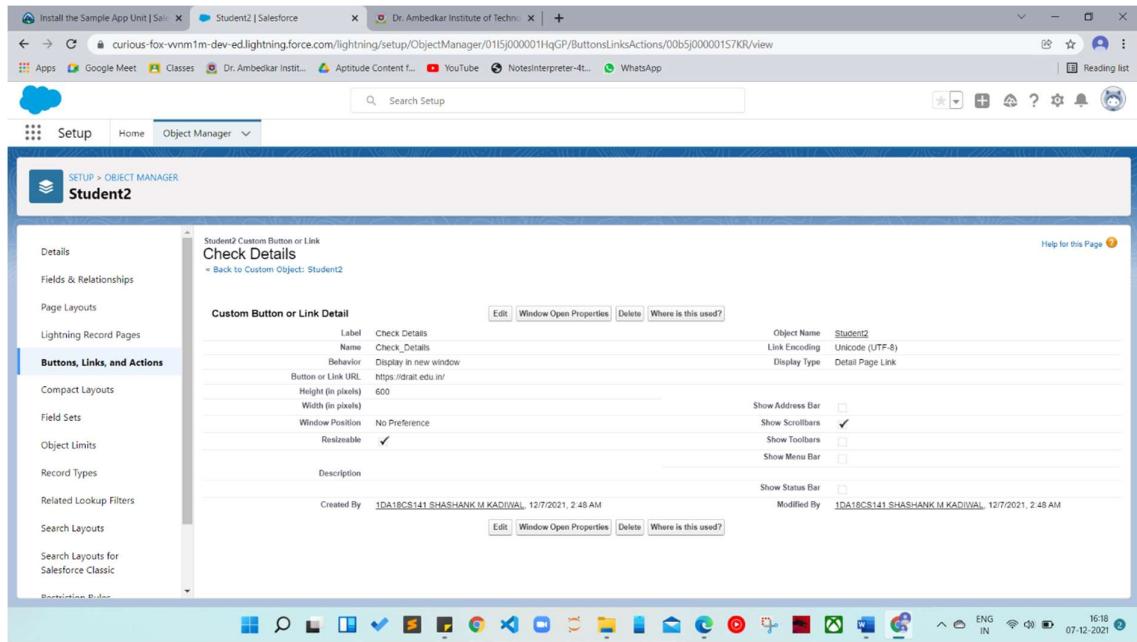
- Field Label: Semester, Data Type: Number (2,0).

The screenshot shows the Salesforce setup interface for the 'Student2' object. The left sidebar includes links for Details, Fields & Relationships, Page Layouts, Lightning Record Pages, Buttons, Links, and Actions, Compact Layouts, Field Sets, Object Limits, Record Types, Related Lookup Filters, Search Layouts, and Search Layouts for Salesforce Classic. The main content area displays a table titled 'Fields & Relationships' with the following data:

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedById	Lookup(User)		
Last Modified By	LastModifiedById	Lookup(User)		
Owner	OwnerId	Lookup(User,Group)		
Semester	Semester__c	Number(2, 0)		✓
Student2 Name	Name	Text(80)		✓
Technical Branch	Technical_Branch__c	Picklist		
Total Attendance	Total_Attendance__c	Percent(3, 2)		
USN	USN__c	Text(10) (Unique Case Sensitive)		✓

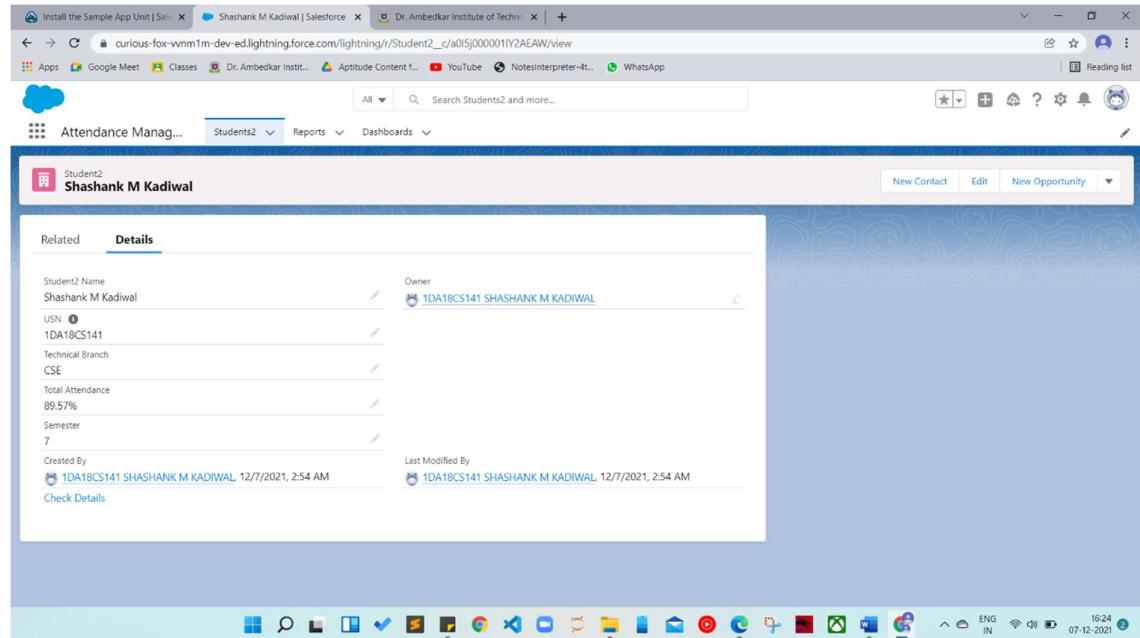
To include Custom Links:

10. Go to “Buttons, Links and Actions” of “Student” Object and click “New Button or Link”.
11. Name it as “Check Details”.
12. Select the radio button “Detail Page Link” as it is a website link.
13. Behaviour: Display in new window.
14. Content Source: URL.
15. Field Type: Student.
16. In the empty space provided, type <http://www.drait.edu.in/> It is a link which redirects to the income tax calculation website.
17. Link Encoding: Unicode (UTF-8).
18. Click Save

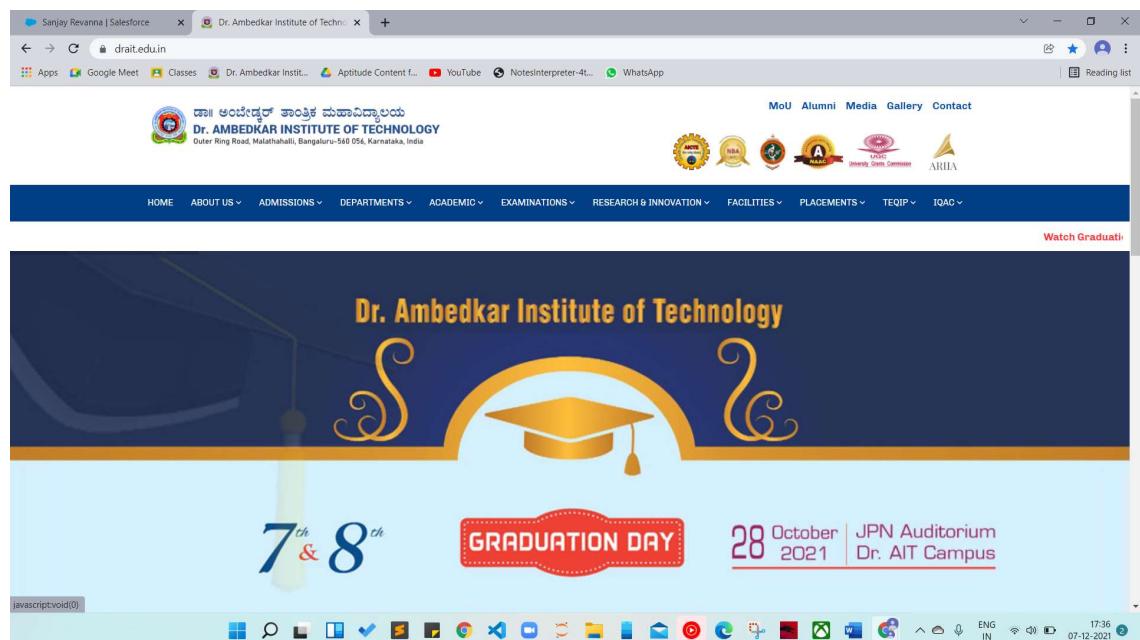


19. Go to Page Layout, Click Student Layout.
20. Click Custom Links, Drag and drop the “Check Details” link in the Custom Link area.
21. Click save.
22. To create an application:
23. Go to “Setup” and type “App Manager” in Quick Find Box.
24. Click on “New Lightning App” to create a Lightning Application.
25. Name it as “Attendance Management”, give the description for your application.
26. Uploading Image and changing colours are optional, then click Next.
27. Navigation Style: Standard Navigation, click Next.
28. No need to add any Utility Bar, click Next.
29. Add the following Items: Student, Records, Reports and Dashboards, click Next.
30. Assign it to System Administrator Profile by selecting System Administrator and pressing right arrow and then click Save & Finish.

31. Go to App Manager, select your application and select Student and click “New” to add some details to your application.



Make Sure you are redirected to the College Website When you click on the Check Details Link.



Reports and Dashboards:

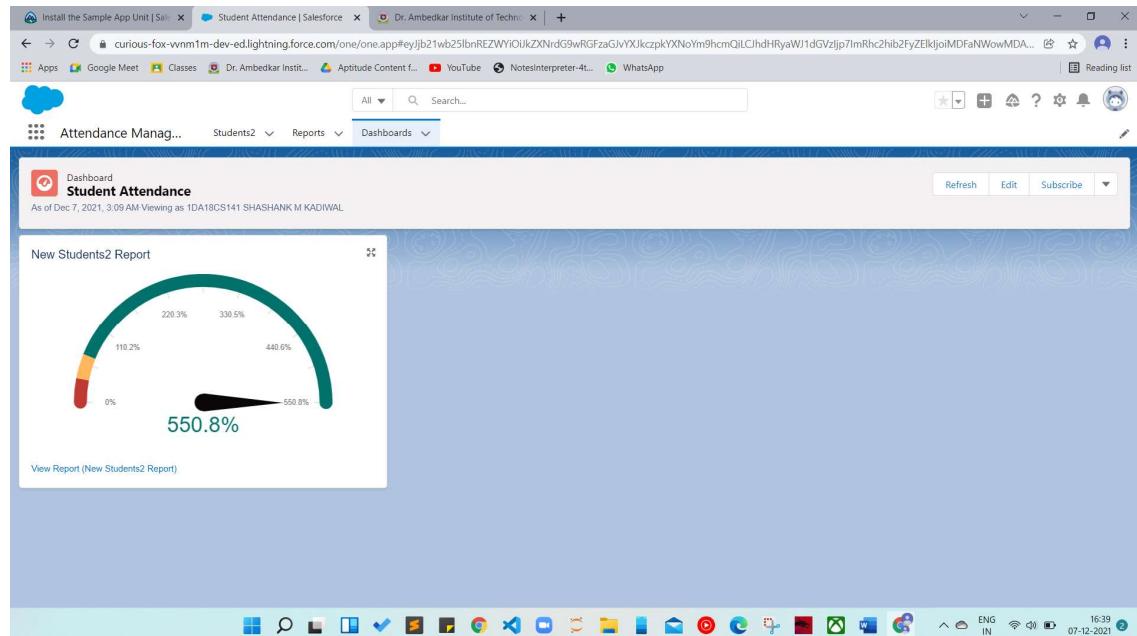
To Create a Student Report:

- Go to “Reports tab” Click on “New Folder” And give it any name then click Save
- Click on “New Report” and from search bar Search for “Attendance Management” and then select it then click Continue.
- Add the required Columns to get the Complete Entered data.
- If you want the report to be grouped by any specific Fields then Search for the field in “Add groups” otherwise it is optional.
- Click on save and name the report as “New Students Report” and then select the folder which you have created.
- Click Save and then Click Run.

Semester	Student2 Name	Technical Branch	USN	Total Attendance
1 (1)	Rahul	EC	1DA21EC101	65.39%
3 (2)	Rakesh V	ISE	1DA20IS176	87.98%
	Ranveer Singh	EEE	1DA20EE120	50.98%
5 (2)	Kiara Advani	TCE	1DA19TC134	78.45%
	Vinod R	CSE	1DA19CS176	78.45%
7 (2)	Sanjay Revanna	CSE	1DA18CS138	99.99%
	Shashank M Kadiwal	CSE	1DA18CS141	89.57%

To Create a Student Dashboard:

- Go to “Dashboard tab” and then click on “New Folder” and give it any Name.
- Click on “New Dashboard” and then name it as “Attendance Management” and select folder that you have created Click on Create.
- Click on the report that you have created and click on that and click select.
- Select any style to represent the data in dashboard.
- Add any filter(s), otherwise it is optional.
- Click on Save and Click Run



9. Create a web application with objects to maintain database of an art gallery which contains objects like artists, arts, inventory and provide a link to any of the art gallery website.

STEPS:

1. Launch your Salesforce Trailhead Playground by opening any module and Switch to Lightning Experience if you are currently in Salesforce Classic by clicking your picture in the right top corner and then click on “Switch to Lightning Experience”.
2. Then go to Setup gear icon and click “Setup”.
3. Click on “Object Manager” and click “Create > Custom Object” to create new Custom Object.
4. Name the object “Artist”.
5. Allow Reports and Allow Search.
6. Check the box in front of “Launch New Custom Tab Wizard after saving this custom object”.
7. To create a Tab for the Object: Select any Tab Style for the object “Artist”. Click Next, Next, leave the defaults and save.
8. To add fields to the Object: Go to “Fields & Relationships” option of Student object and Click “New”.
9. Add the following fields one after the other:
 - Field Label: ID (Length 10), Data Type: Text, provide an example ID as Help Text, make it as required Field, don’t allow Duplicate Values, make it as Case Insensitive and Set this field as the unique record identifier from an external system.
 - Field Label: Art Name and Details, Data Type: Text, Make it as a required field.

- Field Label: Style, Data Type: Picklist,
 - Values: Pencil Sketching, craft design, free hand, Human portrait, drawing, painting etc.
 - Make it as Required Field and restrict values to the values in the picklist

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Art Name and Details	Art_Name_and_Details__c	Text(50)		
Artist Name	Name	Text(80)		✓
Created By	CreatedById	Lookup(User)		
ID	ID__c	Text(10) (External ID) (Unique Case Sensitive)		✓
Last Modified By	LastModifiedById	Lookup(User)		
Owner	OwnerId	Lookup(User/Group)		✓
Style	Style__c	Picklist		

Create one more object to store Collection details:

10. Name the Object “Art”.
11. Allow Reports and Allow Search
12. Check the box in front of “Launch New Custom Tab Wizard after saving this custom object”.
13. Create a Tab for the Object.
14. To add fields to the Object: Go to “Fields & Relationships” option of Student object and Click “New”.
15. Add the following fields one after the other:
 - Field Label: Art Name and Details, Data Type: Master-Detail Relationship, Related to: Artist. Sharing Setting: Read-Only. Leave the defaults and save.

- Field Label: ID (Length 10), Data Type: Text, provide an example ID as Help Text, make it as required Field, don't allow Duplicate Values, make it as Case Insensitive and Set this field as the unique record identifier from an external system.

The screenshot shows the Salesforce Setup interface with the URL <https://curious-fox-vmm1m-dev-ed.lightning.force.com/lightning/setup/ObjectManager/0115j000001HvUJ/FieldsAndRelationships/view>. The page is titled "Art" under "Object Manager". The left sidebar lists options like Details, Fields & Relationships, Page Layouts, Lightning Record Pages, Buttons, Links, and Actions, Compact Layouts, Field Sets, Object Limits, Record Types, Related Lookup Filters, Search Layouts, and Search Layouts for Salesforce Classic. The main content area is titled "Fields & Relationships" and shows a table with the following data:

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Art Name	Name	Text(80)		
Art Name and Details	Art_Name_and_Details__c	Master-Detail(Artist)		
Created By	CreatedById	Lookup(User)		
ID	ID_c	Text(10) (External ID) (Unique Case Sensitive)		
Last Modified By	LastModifiedById	Lookup(User)		

Create one more object to store inventory details:

16. Name the Object “Inventory”.
17. Allow Reports and Allow Search.
18. Check the box in front of “Launch New Custom Tab Wizard after saving this custom object”
19. Create a Tab for the Object.
20. To add fields to the Object: Go to “Fields & Relationships” option of Student object and Click “New”.
21. Add the following fields one after the other:
 - Field Label: Quantity, Data Type: Number. Make it as a required field.

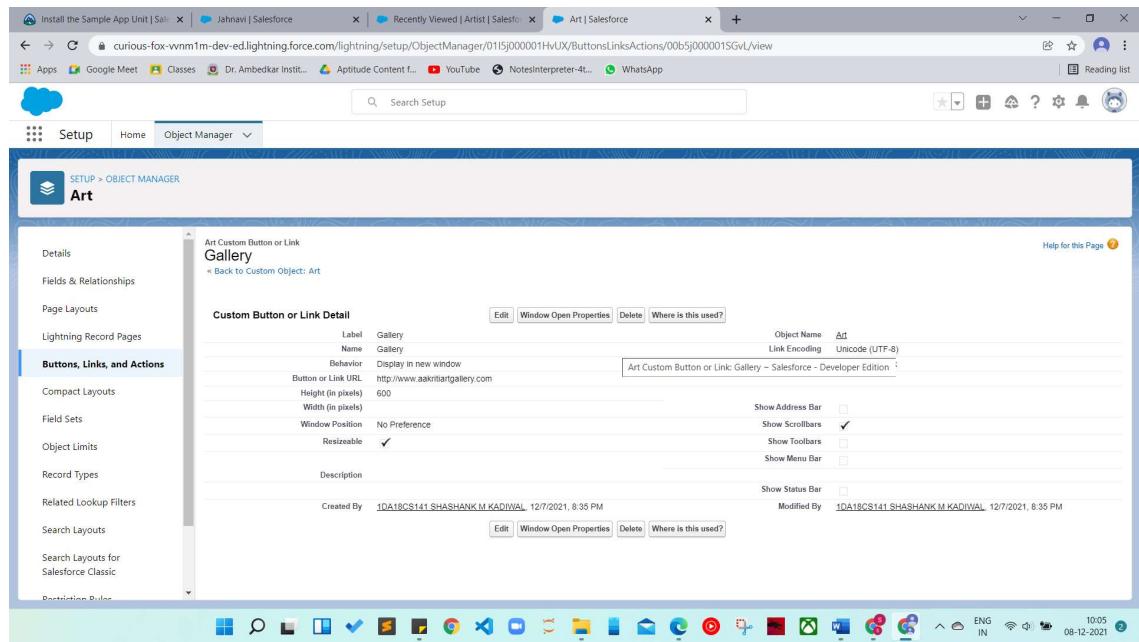
To give a link to any art gallery website:

22. Go to “Buttons, Links and Actions” of Art Object and click “New Button or Link”.

23. Name it as “Gallery”.
24. Select the radio button “Detail Page Link” as it is a website link.
25. Behaviour: Display in new window.
26. Content Source: URL 27. Field Type: Gallery.
28. In the empty space provided, type, <http://www.aakritiartgallery.com/> Link Encoding: Unicode (UTF-8).
30. Click Save.
31. Go to Page Layout, Click Art Layout.
32. Click Custom Links, Drag and drop the “Gallery” link in the Custom Link area.
33. Click save.

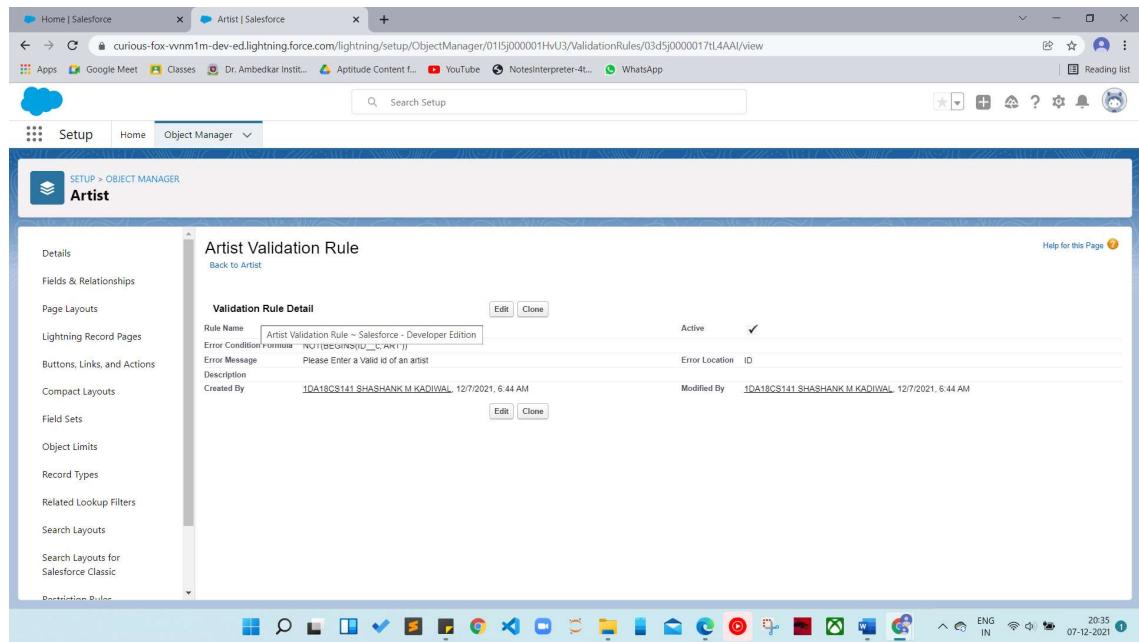
The screenshot shows the Salesforce Setup interface, specifically the Object Manager for the 'Inventory' object. The 'Fields & Relationships' section is displayed, listing five fields:

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedBy	Lookup(User)		
Inventory Name	Name	Text(80)		
Last Modified By	LastModifiedBy	Lookup(User)		
Owner	OwnerId	Lookup(User,Group)		
Quantity	Quantity_c	Number(18, 0)		



To add a rule to the Artist id so that it should take valid id:

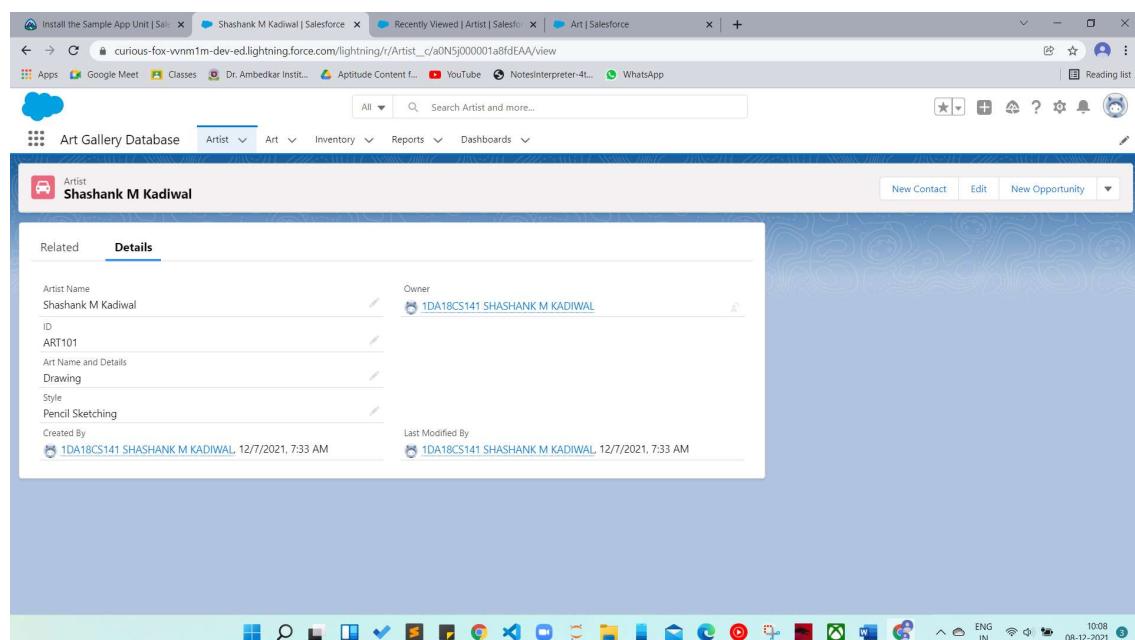
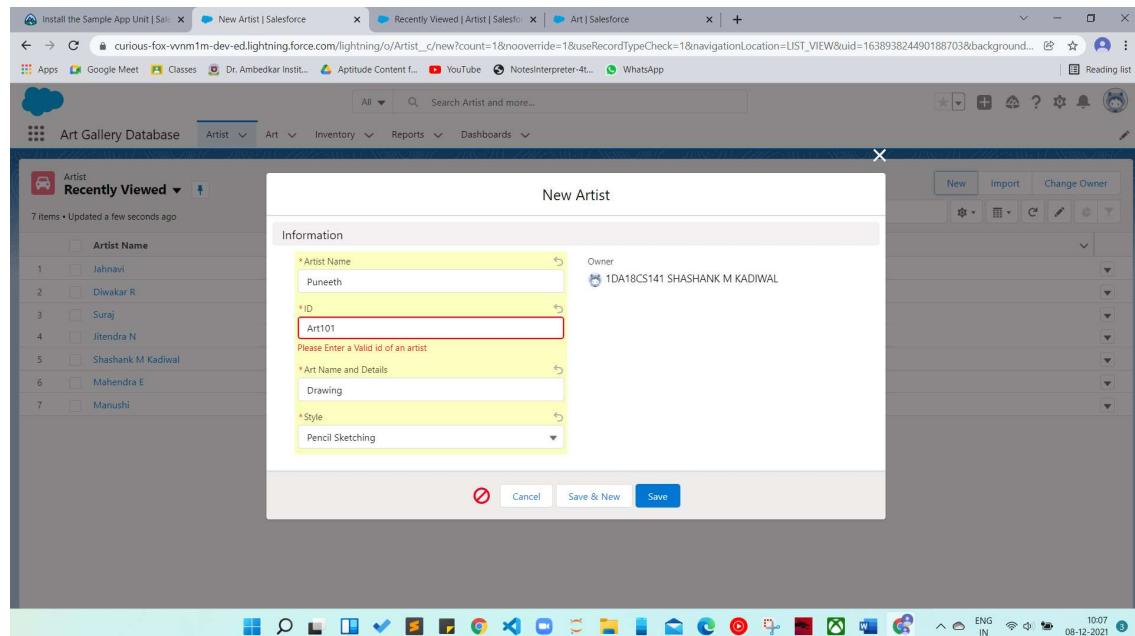
- Go to Validation Rule of Artist Object and click “New”.
- Name it as “Artist id validation”.
- Error Condition Formula: NOT(BEGINS(ID__c,'ART')).
- Error Message: Please Enter a Valid id of an artist.
- Error Location: Field –id.
- Click Save

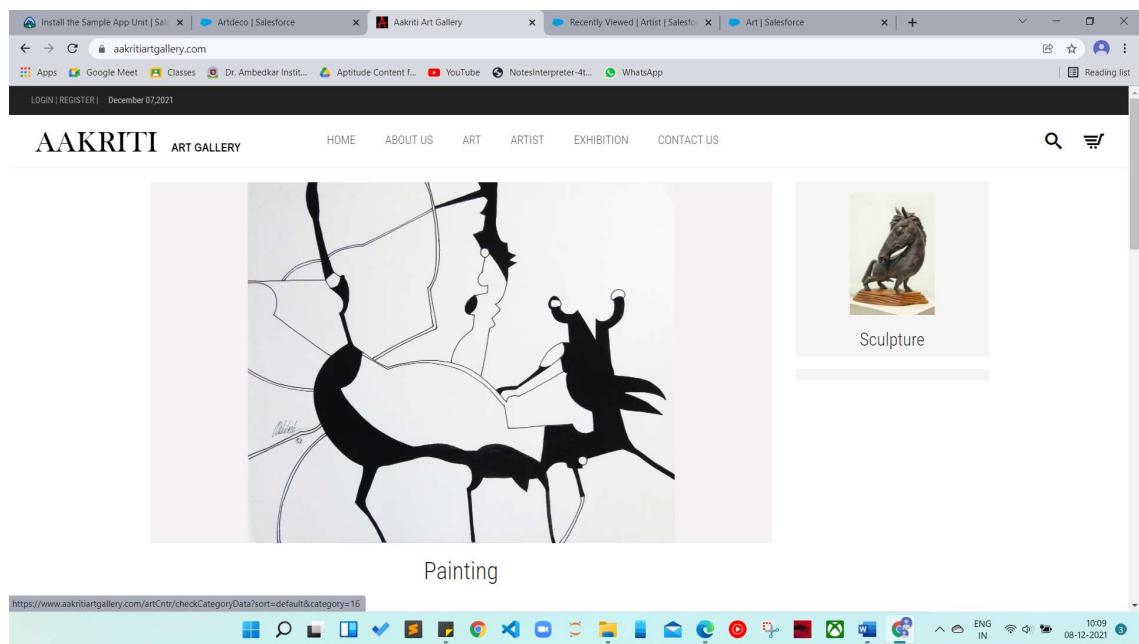
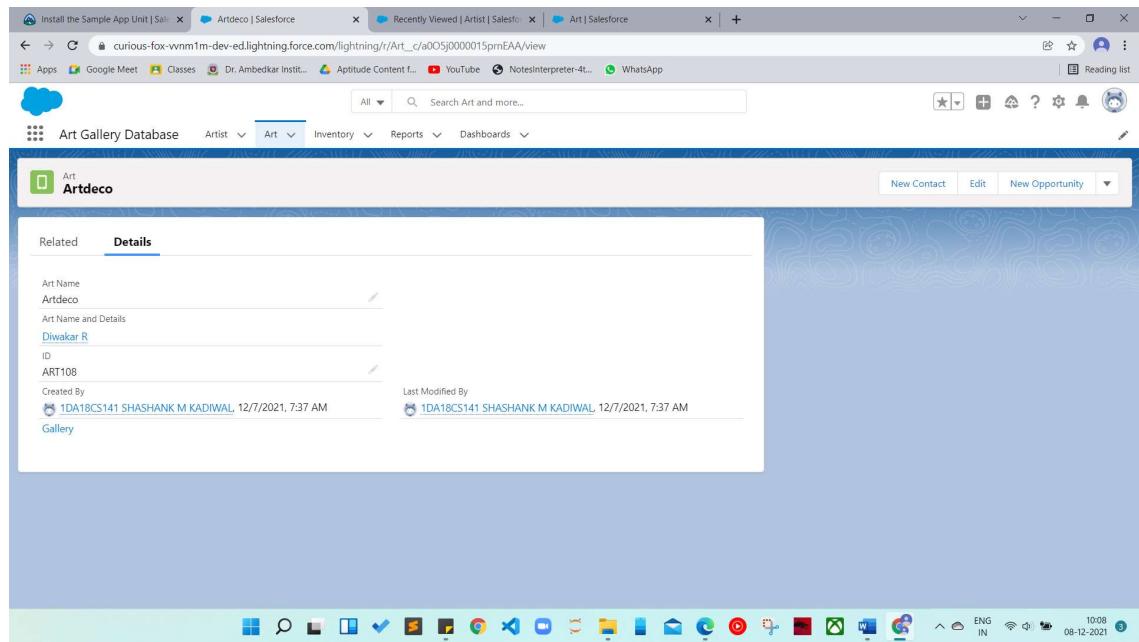


To create an application:

34. Go to “Setup” and type “App Manager” in Quick Find Box.
35. Click on “New Lightning App” to create a Lightning Application.
36. Name it as “Art Gallery Database”, give the description for your application.
37. Uploading Image and changing colours are optional, then click Next.
38. Navigation Style: Standard Navigation, click Next.
39. No need to add any Utility Bar, click Next.
40. Add the following Items: Artists, Arts, Inventories, Reports and Dashboards, click Next.
41. Assign it to System Administrator Profile by selecting System Administrator and pressing right arrow and then click Save & Finish.
42. Go to App Manager, select your application and select Faculties and click “New” to add some details to your application.
43. Click the entry you added, go to details.
44. Press the “Gallery” link to check the details.

45. Click OK so that it will redirect you to the website. Make sure it should an error when an invalid id is given.





Reports and Dashboards:

To Create an Artists Report:

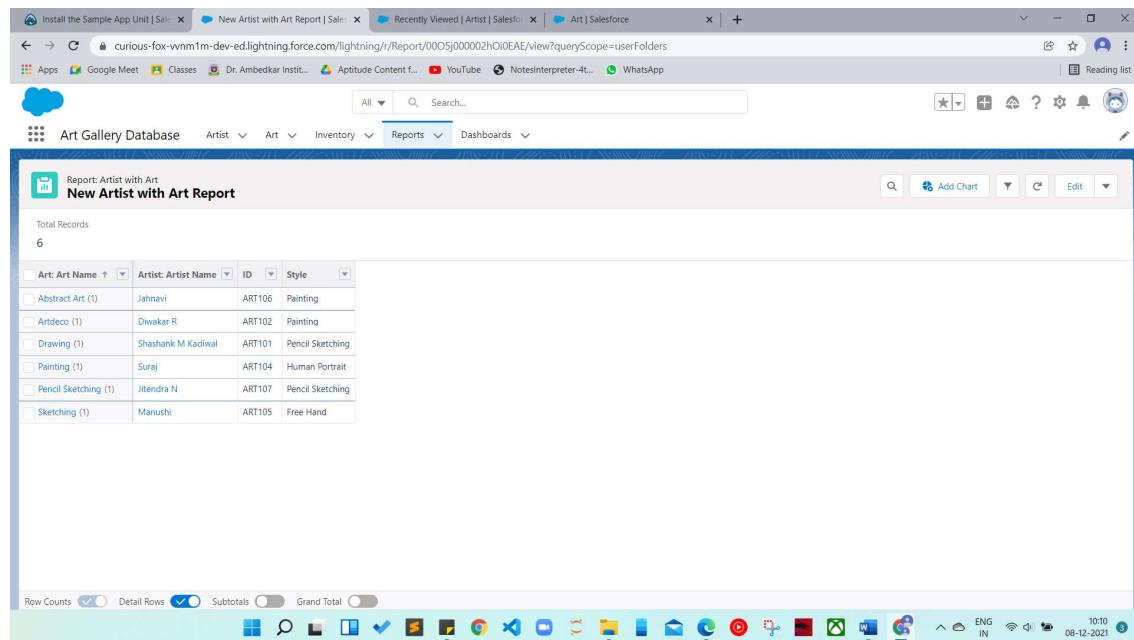
- Go to “Reports tab” Click on “New Folder” And give it any name and Click Save.
- Click on “New Report” and from search bar Search for “Artists” and then select it and then Click Continue.
- Add the required Columns to get the Complete Entered data.
- If you want the report to be grouped by any specific Fields then Search for the field in “Add groups” otherwise it is optional.
- Click on save and name the report as “New Artists Report” and then select the folder which you have created.
- Click Save and then Click Run

The screenshot shows a Salesforce report interface. The title of the report is "Report: Artist New Artist Report". The table displays the following data:

Style	Artist: Artist Name	ID	Art Name and Details
Pencil Sketching (2)	Shashank M Kadiwal	ART101	Drawing
	Jitendra N	ART107	Pencil Sketch
Craft Design (1)	Mahendra E	ART103	Crafting
Free Hand (1)	Manushi	ART105	Sketching
Human Portrait (1)	Suraj	ART104	Human Portrait
Painting (2)	Diwakar R	ART102	Painting
	Jahnavi	ART106	Abstract Art

To Create an Arts Report:

- Go to “Reports tab” Click on “New Folder” And give it any name.
- Click on “New Report” and from search bar Search for “Artists with Arts” and then select it.
- Add the required Columns to get the Complete Entered data.
- If you want the report to be grouped by any specific Fields then Search for the field in “Add groups” otherwise it is optional.
- Click on save and name the report as “New Artists with Arts Report” and then select the folder which you have created.
- Click Save and then Click Run



Art: Art Name	Artist: Artist Name	ID	Style
Abstract Art (1)	Jahnavi	ART106	Painting
Artdeco (1)	Diwakar R	ART102	Painting
Drawing (1)	Shashank M Kadiwal	ART101	Pencil Sketching
Painting (1)	Suraj	ART104	Human Portrait
Pencil Sketching (1)	Jitendra N	ART107	Pencil Sketching
Sketching (1)	Manushi	ART105	Free Hand

To Create an Inventory Report:

- Go to “Reports tab” Click on “New Folder” And give it any name.
- Click on “New Report” and from search bar Search for “Inventories” and then select it.
- Add the required Columns to get the Complete Entered data.
- If you want the report to be grouped by any specific Fields then Search for the field in “Add groups” otherwise it is optional.
- Click on save and name the report as “New Inventories Report” and then select the folder which you have created.
- Click Save and then Click Run.

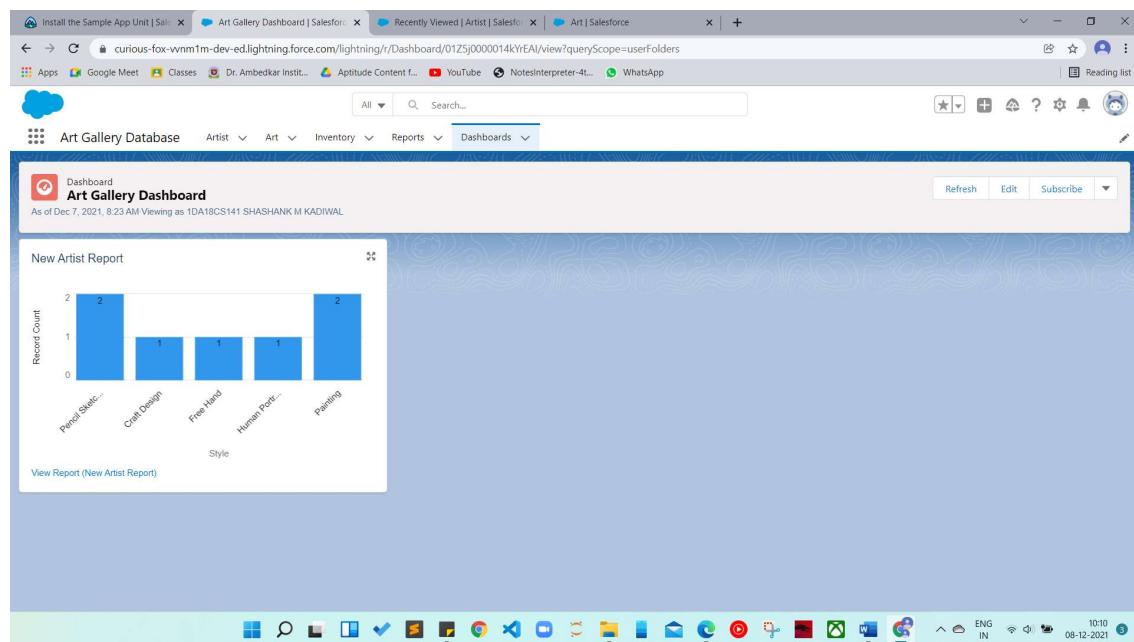
The screenshot shows a Salesforce report interface. The title bar says "Report: Inventory New Inventory Report". The main area displays a table with the following data:

Quantity	Inventory: Inventory Name	Price
1 (1)	Craft Works	\$200
2 (2)	Painting	\$400
	Human Portrait with Pencil Shade	\$400
3 (2)	Landscape Drawing	\$100
	Abstract Art Frames	\$100

At the bottom of the report, there are several buttons: Row Counts, Detail Rows, Subtotals, and Grand Total. The status bar at the bottom right shows the date as 08-12-2021 and the time as 10:10.

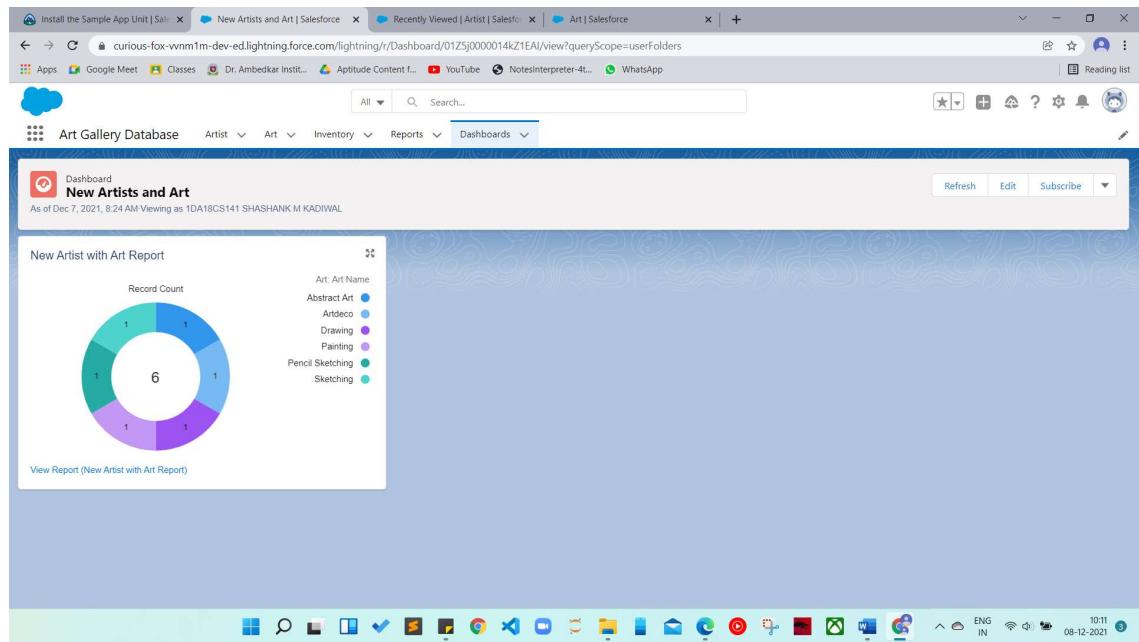
To Create an Artists Dashboard:

- Go to “Dashboard tab” and then click on “New Folder” and give it any Name.
- Click on “New Dashboard” and then name it as “Art Gallery Dashboard” and select folder that you have created Click on Create.
- Click on the report that you have created and click on that and click select.
- Select any style to represent the data in dashboard.
- Add any filter(s), otherwise it is optional. • Click on Save and Click Run



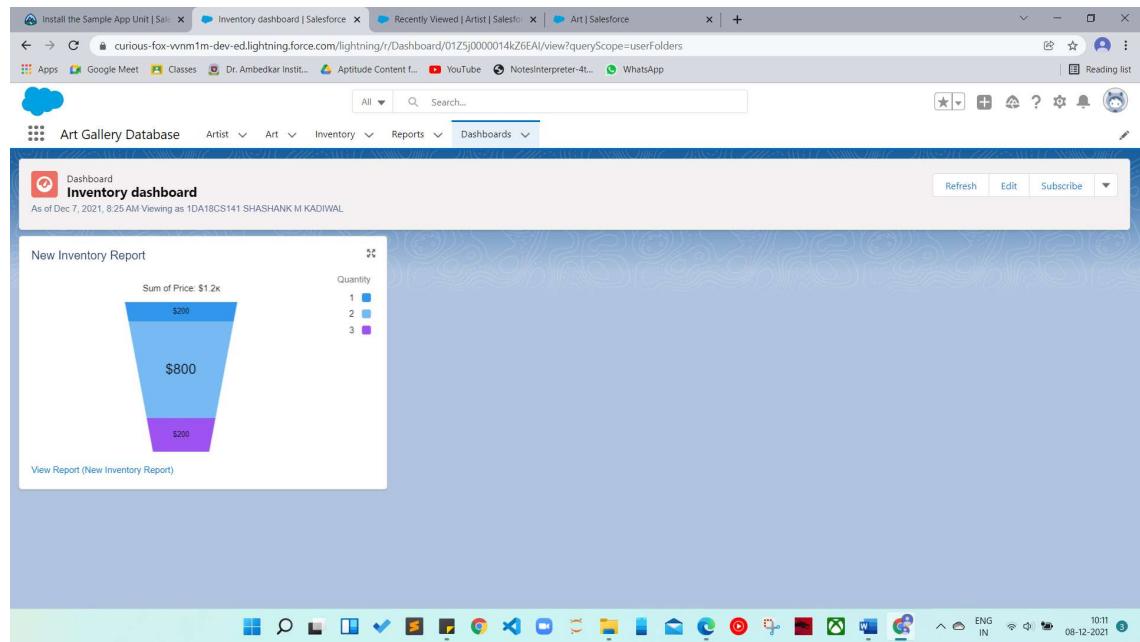
To Create an Arts Dashboard:

- Go to “Dashboard tab” and then click on “New Folder” and give it any Name.
- Click on “New Dashboard” and then name it as “Artists with art Dashboard” and select folder that you have created Click on Create.
- Click on the report that you have created and click on that and click select.
- Select any style to represent the data in dashboard.
- Add any filter(s), otherwise it is optional.
- Click on Save and Click Run



To Create an Inventory Dashboard:

- Go to “Dashboard tab” and then click on “New Folder” and give it any Name.
- Click on “New Dashboard” and then name it as “Inventory Dashboard” and select folder that you have created Click on Create.
- Click on the report that you have created and click on that and click select.
- Select any style to represent the data in dashboard.
- Add any filter(s), otherwise it is optional.
- Click on Save and Click Run



10. Create a web application with objects to record details about staff, syllabus and activities of a department and provide a link to college website from any of the objects.

STEPS:

1. Launch your Salesforce Trailhead Playground by opening any module and Switch to Lightning Experience if you are currently in Salesforce Classic by clicking your picture in the right top corner and then click on “Switch to Lightning Experience”
2. Then go to Setup gear icon and click “Setup”.
3. Click on “Object Manager” and click “Create > Custom Object” to create new Custom Object.
4. Name the object “Staff”.
5. Allow Reports and Allow Search.
6. Check the box in front of “Launch New Custom Tab Wizard after saving this custom object”.
7. To create a Tab for the Object: Select any Tab Style for the object “Staff”. Click Next, Next, leave the defaults and save.
8. To add fields to the Object: Go to “Fields & Relationships” option of Student object and Click “New”.
9. Add the following fields one after the other:
 - Field Label: ID (Length 10), Data Type: Text, provide an example ID as Help Text, make it as Required Field, don’t allow Duplicate Values, make it as Case Insensitive and Set this field as the unique record identifier from an external system.
 - Field Label: Branch, Data Type: Text, Make it as a required field.

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Branch	Branch_c	Text(50)		
Created By	CreatedById	Lookup(User)		
ID	ID_c	Text(20) (External ID) (Unique Case Sensitive)		
Last Modified By	LastModifiedById	Lookup(User)		
Owner	OwnerId	Lookup(User/Group)		
Staff Name	Name	Text(80)		

Create one more object to store Syllabus details:

10. Name the Object “Syllabus”.
11. Allow Reports and Allow Search.
12. Check the box in front of “Launch New Custom Tab Wizard after saving this custom object”
13. Create a Tab for the Object.
14. To add fields to the Object:

Go to “Fields & Relationships” option of Student object and Click “New”.

15. Add the following fields one after the other:
 - Field Label: Subject Code, Data Type: Text.
 - Field Label: Credits, Data Type: Number.

Create one more object to store department activities’ details:

16. Name the Object “Activities”.
17. Allow Reports and Allow Search.

18. Check the box in front of “Launch New Custom Tab Wizard after saving this custom object”

19. Create a Tab for the Object.

20. To add fields to the Object:

Go to “Fields & Relationships” option of Student object and Click “New”.

21. Add the following fields one after the other:

- Field Label: Details, Data Type: Text. Make it as a required field

The screenshot shows the Salesforce Object Manager interface. The left sidebar has a tree view with 'Syllabus' selected. Under 'Fields & Relationships', several options like Page Layouts, Lightning Record Pages, and Buttons, Links, and Actions are listed. The main content area is titled 'Fields & Relationships' and shows a table of existing fields for the 'Syllabus' object. The table includes columns for FIELD LABEL, FIELD NAME, DATA TYPE, CONTROLLING FIELD, and INDEXED. The fields listed are: Created By (CreatedBy), Credits (Credits__c), Last Modified By (LastModifiedBy), Owner (OwnerId), Subject Code (Subject_Code__c), and Syllabus Name (Name). The 'INDEXED' column for the 'Name' field has a checkmark. At the top of the table, there are buttons for Quick Find, New, Deleted Fields, Field Dependencies, and Set History Tracking.

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedBy	Lookup(User)		
Credits	Credits__c	Number(18, 0) (Unique)		
Last Modified By	LastModifiedBy	Lookup(User)		
Owner	OwnerId	Lookup(User,Group)		
Subject Code	Subject_Code__c	Text(50)		
Syllabus Name	Name	Text(80)		✓

The screenshot shows the Salesforce Setup interface with the URL curious-fox-vnvm1m-dev-ed.lightning.force.com/lightning/setup/ObjectManager/0115j000001HsDP/FieldsAndRelationships/view. The page title is "Activities". On the left, there's a sidebar with links like Details, Fields & Relationships, Page Layouts, Lightning Record Pages, Buttons, Links, and Actions, Compact Layouts, Field Sets, Object Limits, Record Types, Related Lookup Filters, Search Layouts, and Search Layouts for Salesforce Classic. The main content area is titled "Fields & Relationships" and shows a table with 5 items, sorted by Field Label. The table columns are FIELD LABEL, FIELD NAME, DATA TYPE, CONTROLLING FIELD, and INDEXED. The data is as follows:

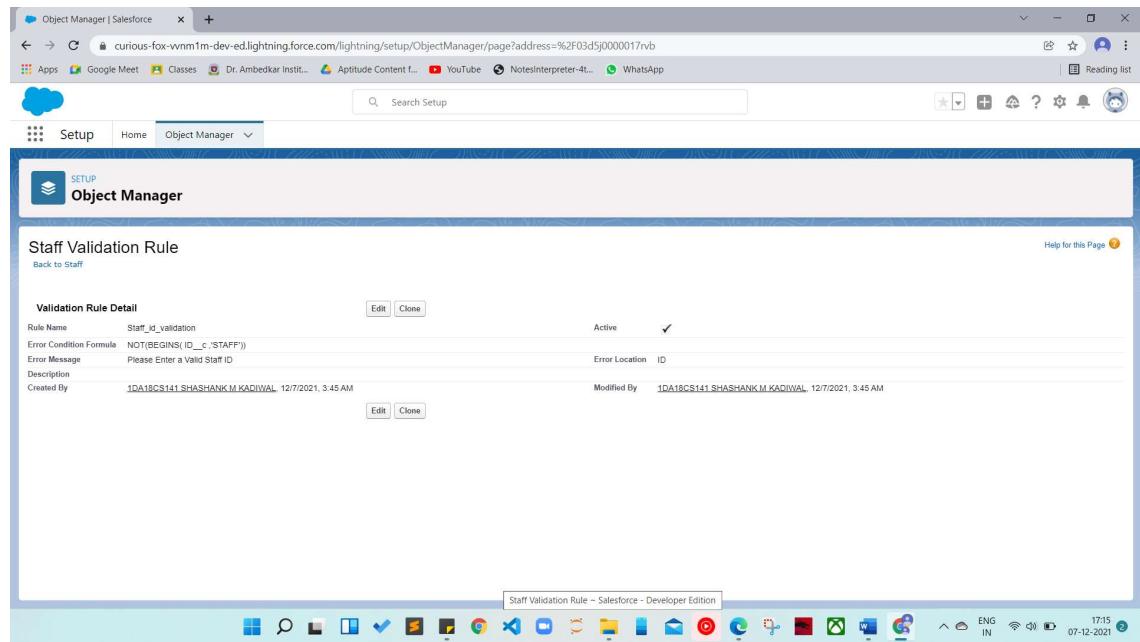
FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Activities Name	Name	Text(80)		✓
Created By	CreatedBy	Lookup(User)		
Details	Details__c	Text(50)		
Last Modified By	LastModifiedBy	Lookup(User)		
Owner	OwnerId	Lookup(User,Group)		✓

Validation Rules:

1. Staff id Validation:

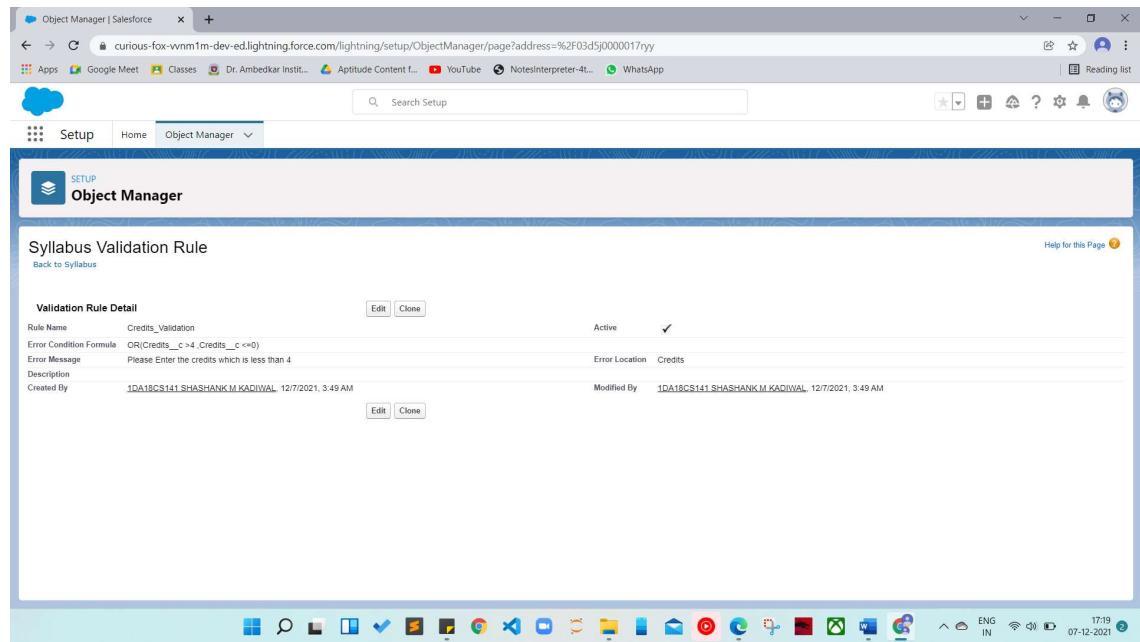
To add a rule to the Staff is so that it should take only valid ids:

- Go to Validation Rule of Staff Object and click “New”
- Name it as “Staff id validation”.
- Error Condition Formula: NOT(BEGINS(ID__c,'STAFF')).
- Error Message: Please enter a valid staff Id.
- Error Location: Field – ID.
- Click Save.



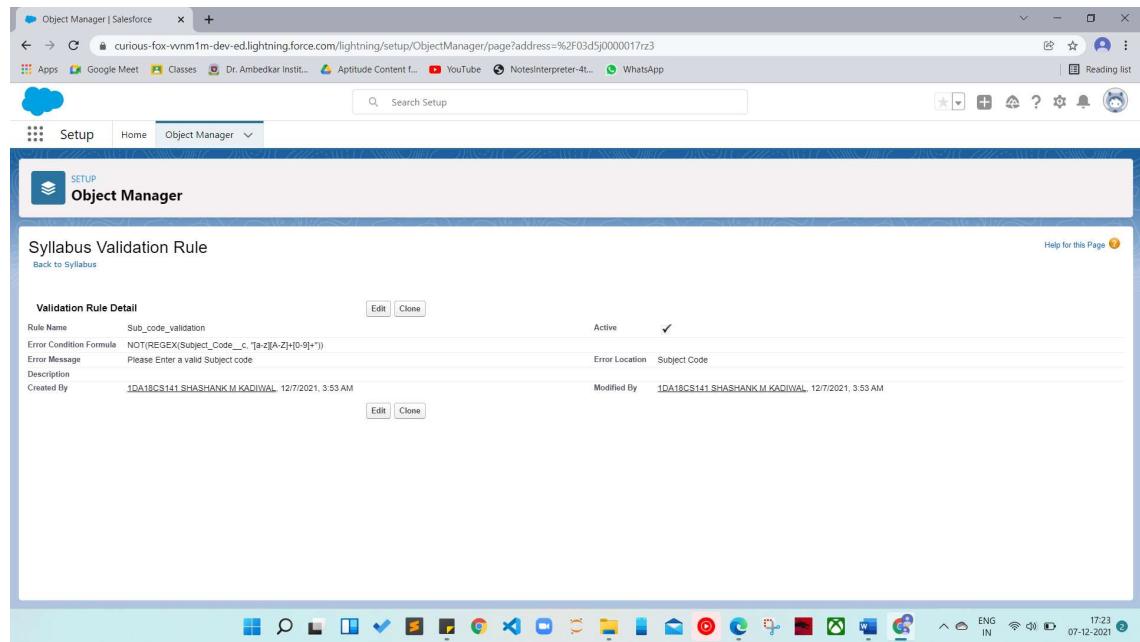
To add a rule to the Subject code so that it should not take invalid sub code:

- Go to Validation Rule of Syllabus Object and click “New”.
- Name it as “Sub code validation”
- Error Condition Formula: NOT (REGEX (Subject_Code__c, “[0-9]{2}+[a-z][A-Z]{2}+[0-9]{2}+”)).
- Error Message: Please Enter a valid Subject code.
- Error Location: Field – Subject Code.
- Click Save



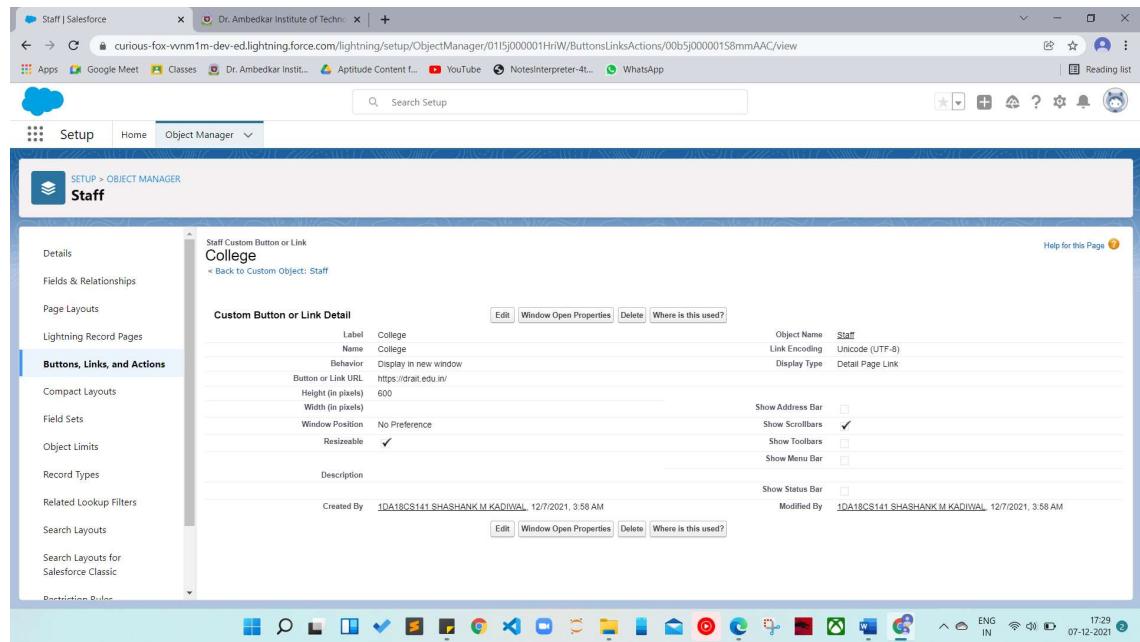
To add a rule to the Credits so that it Should not take more than 4 credits:

- Go to Validation Rule of Syllabus Object and click “New”.
- Name it as “Credits validation”.
- Error Condition Formula: OR (credits__c >4, Credits__c <=0).
- Error Message: Please Enter the credits which is less than 4.
- Error Location: Field –Credit.
- Click Save.



To give a link to college website:

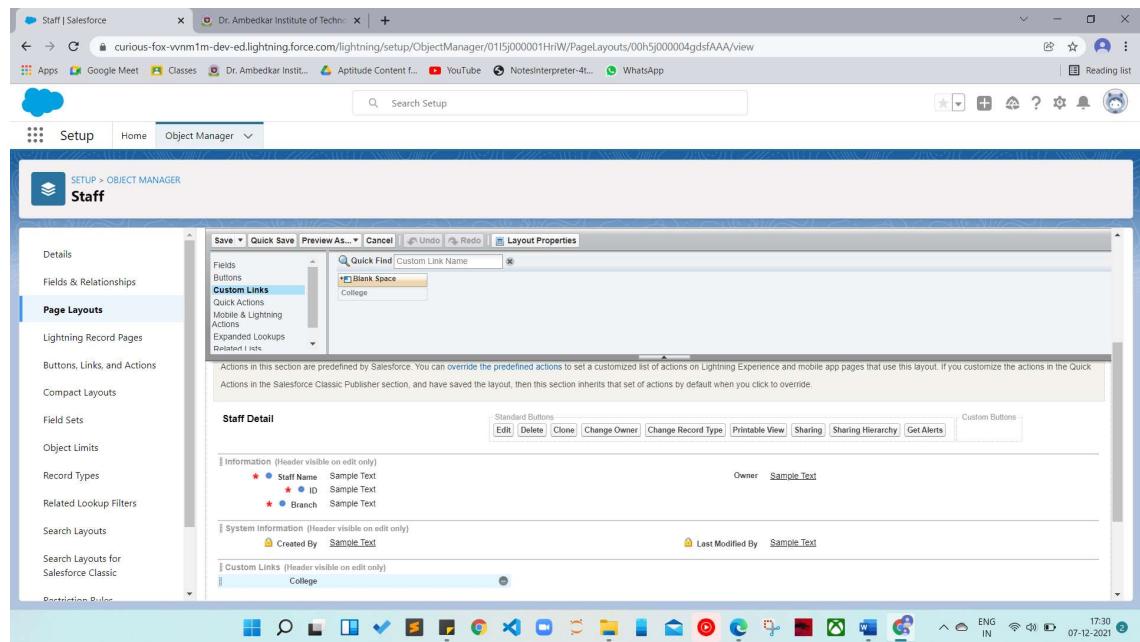
22. Go to “Buttons, Links and Actions” of Art Object and click “New Button or Link”.
23. Name it as “College”.
24. Select the radio button “Detail Page Link” as it is a website link.
25. Behaviour: Display in new window.
26. Content Source: URL.
27. Field Type: College.
28. In the empty space provided, type <https://www.drait.edu.in/>
29. Link Encoding: Unicode (UTF-8).
30. Click Save



31. Go to Page Layout, Click Activities Layout.

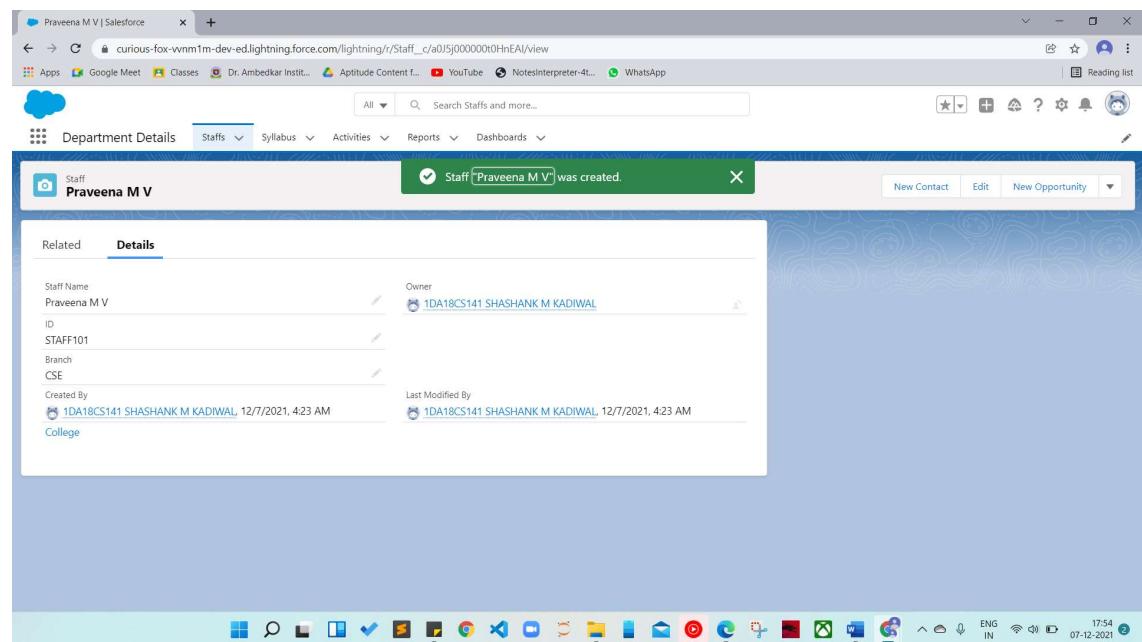
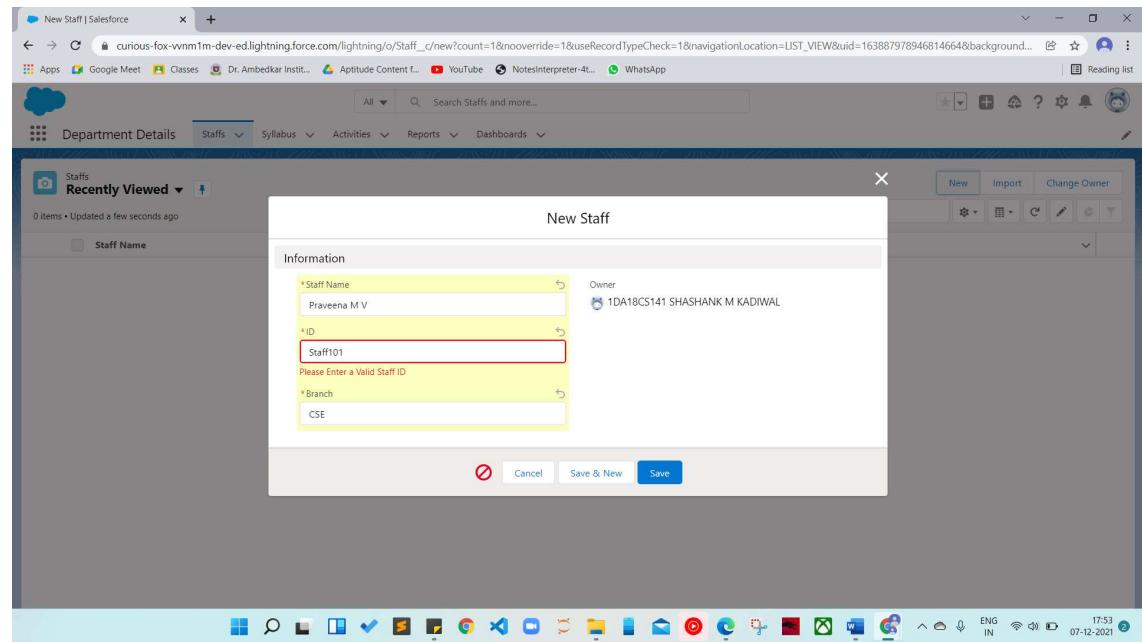
32. Click Custom Links, Drag and drop the “College” link in the Custom Link area.

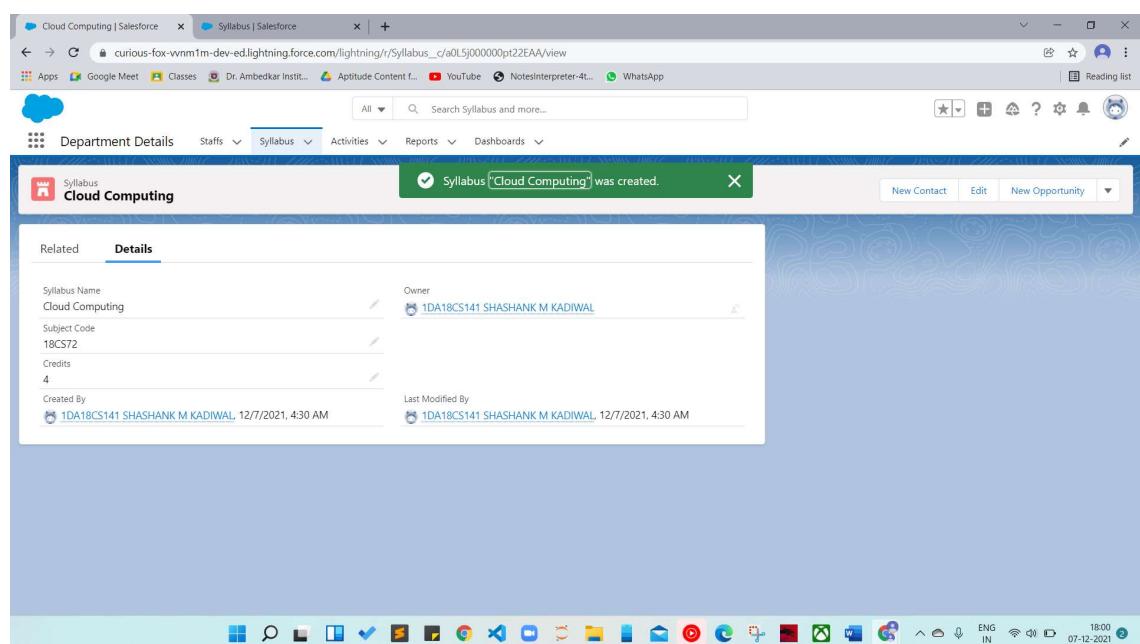
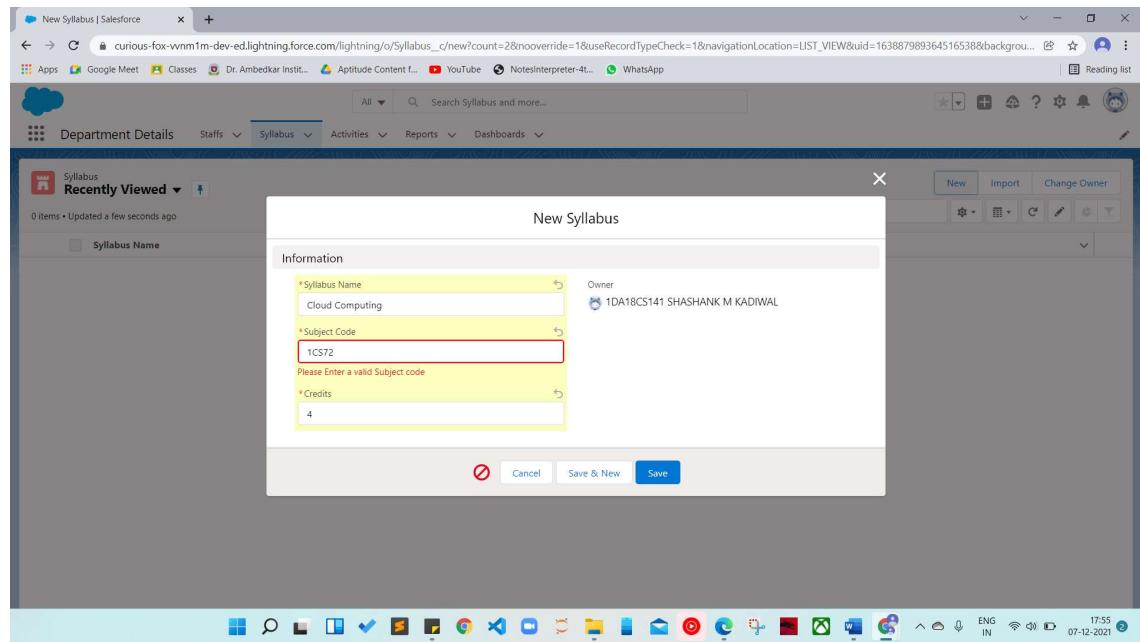
33. Click Save.

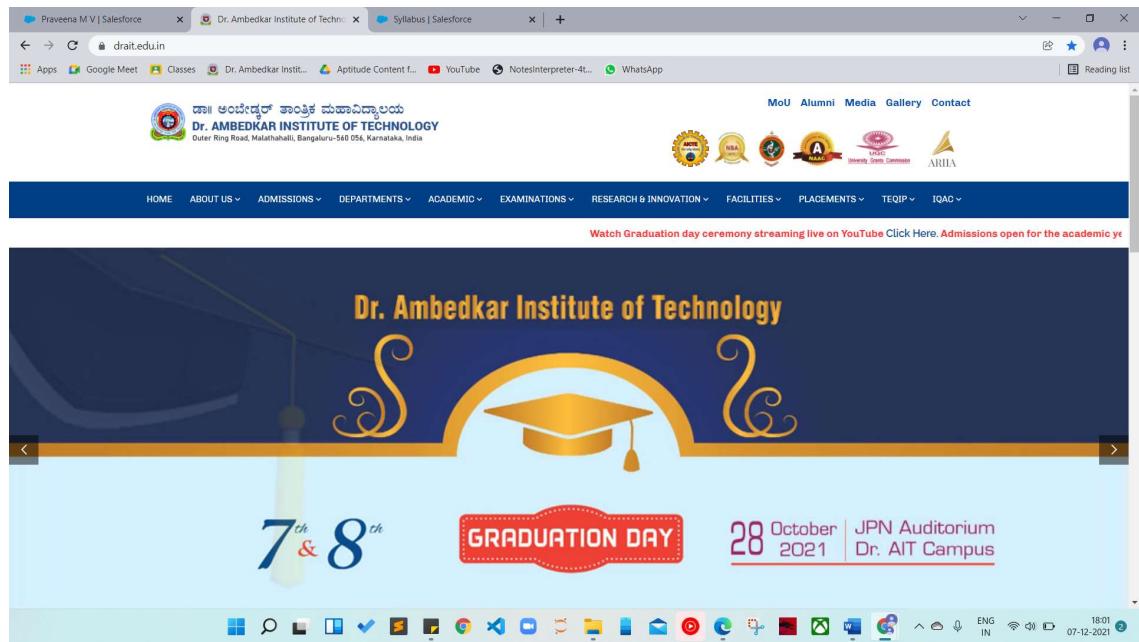


To create an application:

34. Go to “Setup” and type “App Manager” in Quick Find Box.
35. Click on “New Lightning App” to create a Lightning Application.
36. Name it as “Department Details”, give the description for your application.
37. Uploading Image and changing colours are optional, then click Next.
38. Navigation Style: Standard Navigation, click Next.
39. No need to add any Utility Bar, click Next.
40. Add the following:
 41. Items: Staff, Syllabuses, Activities, Reports and Dashboards, click Next.
 42. Assign it to System Administrator Profile by selecting System Administrator and pressing right arrow and then click Save & Finish.
 43. Go to App Manager, select your application and select Faculties and click “New” to add some details to your application.
 44. Click the entry you added, go to details.
 45. Press the “College” link to check the details.
 46. Click OK so that it will redirect you to the website. Make Sure You will get an error when u give invalid staff id, credits and Subject code







Reports and Dashboards:

To Create a Staff Report:

- Go to “Reports tab” Click on “New Folder” And give it any name then select Save.
- Click on “New Report” and from search bar Search for “Staffs” and then select it then select Continue.
- Add the required Columns to get the Complete Entered data.
- If you want the report to be grouped by any specific Fields then Search for the field in “Add groups” otherwise it is optional.
- Click on save and name the report as “New Staffs Report” and then select the folder which you have created.
- Click Save and then Click Run.

The screenshot shows a Salesforce report titled "Report: Staffs New Staffs Report". The report displays a list of staff members with their names and IDs, grouped by their respective branches. The data is presented in a table with columns for Branch, Staff: Staff Name, and ID.

Branch	Staff: Staff Name	ID
Civil (1)	Bhanumati N	STAFF104
CSE (1)	Praveena M V	STAFF101
ECE (1)	Shalini G	STAFF102
EEE (1)	Shamitha S	STAFF105
ISE (1)	Shamanth D	STAFF106
Mechanical (2)	Manjunath M	STAFF108
	Murali A	STAFF107

Below the table, there are several filter and sorting options: Row Counts, Detail Rows, Subtotals, and Grand Total. At the bottom of the screen, a taskbar is visible with various application icons.

To Create a Syllabus Report:

- Go to “Reports tab” Click on “New Folder” And give it any name then select Save.
- Click on “New Report” and from search bar Search for “Syllabus” and then select it then select Continue.
- Add the required Columns to get the Complete Entered data.
- If you want the report to be grouped by any specific Fields then Search for the field in “Add groups” otherwise it is optional.
- Click on save and name the report as “New Syllabus Report” and then select the folder which you have created.
- Click Save and then Click Run

The screenshot shows a Salesforce Lightning interface with three tabs open: 'New Syllabus Report | Salesforce', 'Dr. Ambedkar Institute of Techn...', and 'Syllabus | Salesforce'. The main content area displays a report titled 'Report: Syllabus New Syllabus Report'. The report shows a table with 5 total records. The columns are 'Credits ↑', 'Syllabus: Syllabus Name ↓', and 'Subject Code ↓'. The data rows are:

Credits ↑	Syllabus: Syllabus Name ↓	Subject Code ↓
2 (1)	Android Lab	18CS76
3 (2)	Software Engineering Operating System	18CS51 18CS35
4 (2)	Cloud Computing Python Programming	18CS72 18CS34

At the bottom of the report, there are buttons for 'Row Counts', 'Detail Rows', 'Subtotals', and 'Grand Total'. Below the report is a standard Windows taskbar with various icons.

To Create an Activities Report:

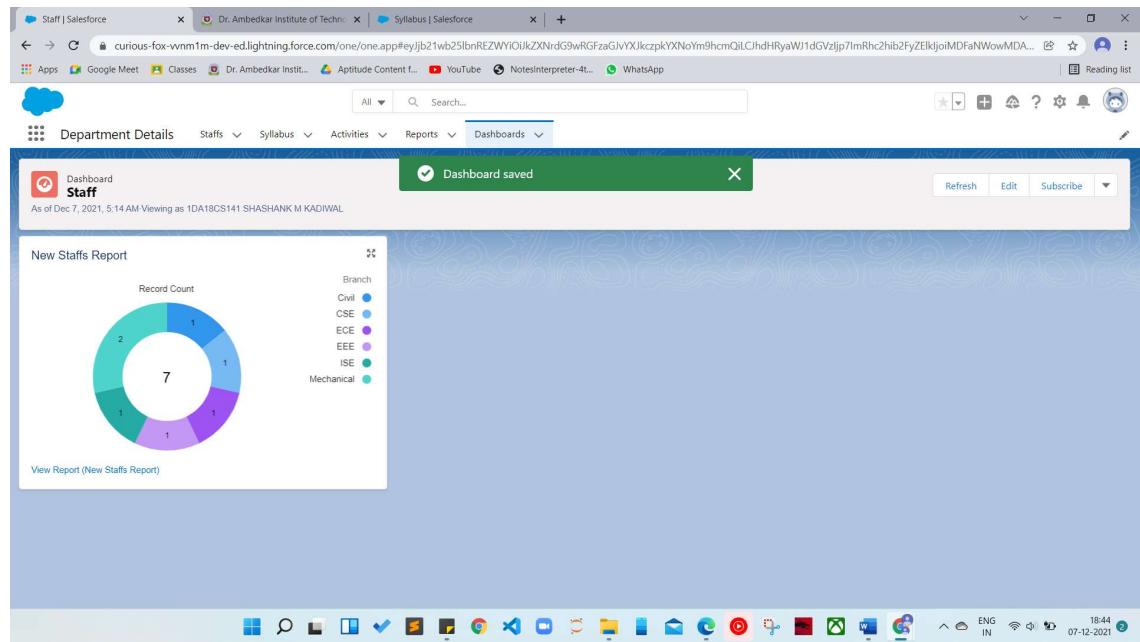
- Go to “Reports tab” Click on “New Folder” And give it any name then select Save.
- Click on “New Report” and from search bar Search for “Activities” and then select it then select Continue.
- Add the required Columns to get the Complete Entered data.
- If you want the report to be grouped by any specific Fields then Search for the field in “Add groups” otherwise it is optional.
- Click on save and name the report as “New Activities Report” and then select the folder which you have created.
- Click Save and then Click Run

The screenshot shows a Salesforce Lightning interface. At the top, there are three tabs: 'New Activities Report | Salesforce', 'Dr. Ambedkar Institute of Techn...', and 'Syllabus | Salesforce'. The main content area is titled 'Report: Activities New Activities Report'. It displays a table with 5 total records. The columns are 'Activities' and 'Details'. The data is as follows:

	Activities	Details
1	Coding	the process or activity of writing code
2	Team Leading	A Management Event
3	Quiz	A test of knowledge
4	Technical Quiz	A Quiz on Tech
5	App Building	Competition to Build App

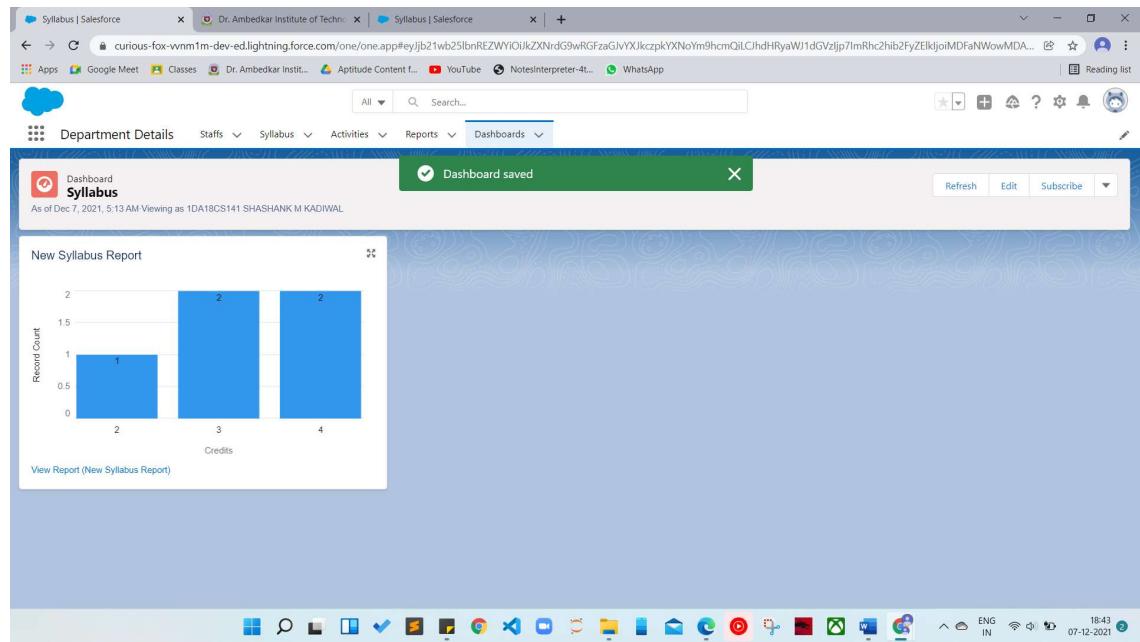
To Create a Staff Dashboard:

- Go to “Dashboard tab” and then click on “New Folder” and give it any Name.
- Click on “New Dashboard” and then name it as “Staff dashboard” and select folder that you have created Click on Create.
- Click on the report that you have created and click on that and click select.
- Select any style to represent the data in dashboard.
- Add any filter(s), otherwise it is optional.
- Click on Save and Click Run



To Create a Syllabus Dashboard:

- Go to “Dashboard tab” and then click on “New Folder” and give it any Name.
- Click on “New Dashboard” and then name it as “Syllabus dashboard” and select folder that you have created Click on Create.
- Click on the report that you have created and click on that and click select.
- Select any style to represent the data in dashboard.
- Add any filter(s), otherwise it is optional.
- Click on Save and Click Run.



To Create an Activities Dashboard:

- Go to “Dashboard tab” and then click on “New Folder” and give it any Name.
- Click on “New Dashboard” and then name it as “Activities dashboard” and select folder that you have created Click on Create.
- Click on the report that you have created and click on that and click select.
- Select any style to represent the data in dashboard.
- Add any filter(s), otherwise it is optional.
- Click on Save and Click Run

The screenshot shows a Salesforce Lightning interface. At the top, there are three tabs: 'Activities | Salesforce', 'Dr. Ambedkar Institute of Techn...', and 'Syllabus | Salesforce'. The main header includes a cloud icon, a search bar, and various navigation links like 'Apps', 'Google Meet', 'Classes', 'Dr. Ambedkar Instit...', 'Aptitude Content ...', 'YouTube', 'NotesInterpreter-4...', 'WhatsApp', and 'Reading list'. Below the header, the page title is 'Department Details' under 'Activities'. A green success message 'Dashboard saved' with a checkmark is displayed. The main content area is titled 'New Activities Report' and contains a table with the following data:

Activities: Activities Name ↑	Details
App Building	Competition to Build App
Coding	the process or activity of writing code
Quiz	A test of knowledge
Team Leading	A Management Event
Technical Quiz	A Quiz on Tech

At the bottom of the report table, there is a link 'View Report (New Activities Report)'. The status bar at the bottom right shows the date '07-12-2021', time '18:45', and system information 'ENG IN'.

CLOUD ANALYST

Cloud Analyst

Cloud Analyst is a tool developed at the University of Melbourne whose goal is to support evaluation of social networks tools according to geographic distribution of users and data centers. In this tool, communities of users and data centers supporting the social networks are characterized and, based on their location; parameters such as user experience while using the social network application and load on the data centre are obtained/logged.



Cloud Analyst is developed by Bhathiya Wickremasinghe et al. at the CLOUDS Laboratory. It is built on top of CloudSim and separates the simulation experimentation from a programming task enabling one to concentrate on the simulation parameters rather than the technicalities of programming. Simulation in Cloud Analyst involves the following steps:

- i. Defining and configuration of User Bases.
- ii. Defining and configuring Data Centers
- iii. Allocating of Virtual Machines in Data Centers.
- iv. Review and Adjustment of various other parameters such as Packet size, Number of packets, Bandwidth, and Load balancing policies.

The Cloud Analyst enables us to model different scenarios of CSPs and User Bases, and provides a comprehensive output detailing the response time, Data Center processing time and total cost involved in the communication and computation.

Installing and Running Cloud Analyst:

1.Download CloudAnalyst

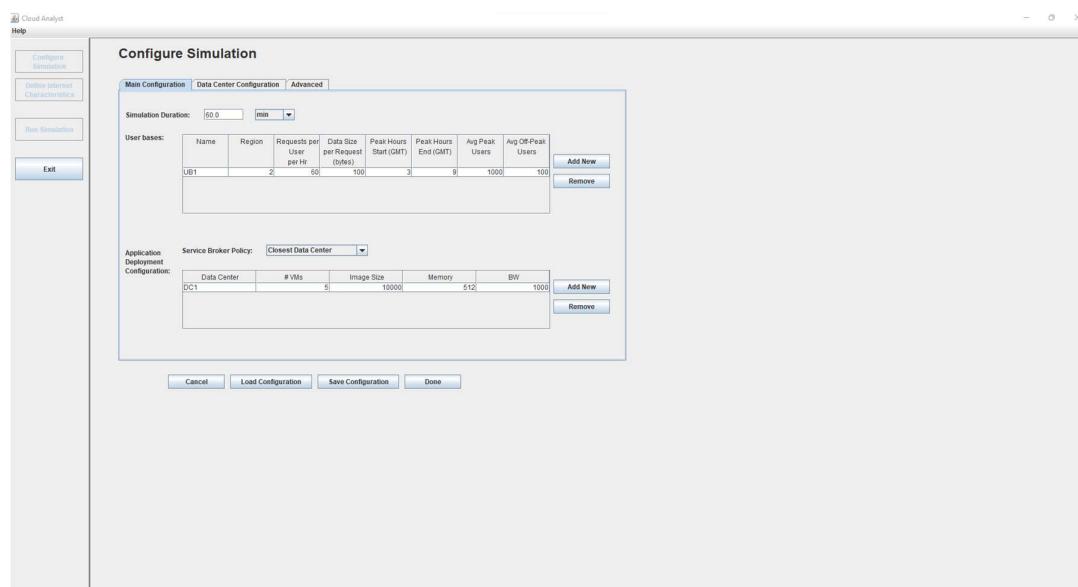
2.Extract the files from the zip file which will give following folder structure.

3.Click on run.bat file.

4.The user can then configure the simulation which includes

- Simulation Duration
- Number of User Bases
- Service broker policy
- Data Center Configuration
- Load Balancing Policy

5. To run the simulation, click on Run Simulation.



Cloud Analyst

Configure Simulation

Main Configuration | Data Center Configuration | Advanced

Data Centers:

Name	Region	Arch	OS	VM	Cost per VM \$/hr	Memory Cost \$/s	Storage Cost \$/Gb	Data Transfer Cost \$/Gb	Physical HW Units
DC1	0x86	Linux	Xen	1	0.1	0.005	0.1	0.1	2

Add New | Remove

Cancel | Load Configuration | Save Configuration | Done

Cloud Analyst

Configure Simulation

Main Configuration | Data Center Configuration | Advanced

User grouping factor in User Bases: (Equivalent to number of simultaneous users from a single user base)

Request grouping factor in Data Centers: (Equivalent to number of simultaneous requests a single application server instance can support)

Executable instruction length per request: [bytes]

Load balancing policy across VM's in a single Data Center:

Cancel | Load Configuration | Save Configuration | Done

Cloud Analyst Simulations

<p>1</p>	<p>Use the following userbase configuration to simulate following scenarios for the given data centre and virtual machine configuration and answer to the following questions.</p> <p>Scenario-1: Nearest data centre with round robin policies.</p> <p>Scenario-2: Optimize response time with round robin policies.</p>																																										
	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>User Base</th><th>Region</th><th>Data Centre</th><th>Peak Hour Users</th><th>Off Peak Hour Users</th><th>Virtual Machines</th></tr> </thead> <tbody> <tr> <td>UB1</td><td>North America</td><td></td><td>1000</td><td>500</td><td rowspan="6" style="text-align: center;">DC1-50</td></tr> <tr> <td>UB2</td><td>South America</td><td></td><td>1200</td><td>800</td><td></td></tr> <tr> <td>UB3</td><td>Europe</td><td>DC1</td><td>2000</td><td>1000</td><td></td></tr> <tr> <td>UB4</td><td>Africa</td><td></td><td>500</td><td>300</td><td></td></tr> <tr> <td>UB5</td><td>Asia</td><td></td><td>3000</td><td>300</td><td></td></tr> <tr> <td>UB6</td><td>Oceania</td><td></td><td>1500</td><td>150</td><td></td></tr> </tbody> </table> <p>i) Tabulate the overall response time of all the scenarios and plot a line graph. ii) Plot a bar graph for the data processing time of all the scenarios. iii) Compare average response time by regions of all scenarios by plotting the line graph. iv) Using Pie Chart show the total cost spent for each scenario.</p>	User Base	Region	Data Centre	Peak Hour Users	Off Peak Hour Users	Virtual Machines	UB1	North America		1000	500	DC1-50	UB2	South America		1200	800		UB3	Europe	DC1	2000	1000		UB4	Africa		500	300		UB5	Asia		3000	300		UB6	Oceania		1500	150	
User Base	Region	Data Centre	Peak Hour Users	Off Peak Hour Users	Virtual Machines																																						
UB1	North America		1000	500	DC1-50																																						
UB2	South America		1200	800																																							
UB3	Europe	DC1	2000	1000																																							
UB4	Africa		500	300																																							
UB5	Asia		3000	300																																							
UB6	Oceania		1500	150																																							
<p>2</p>	<p>Simulate the following scenarios for the given userbase, data centre and virtual machine configuration and answer to the given questions.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Scenario</th><th>Scenario Description</th><th>Load balancing algorithm</th><th>Service Broker Policy</th></tr> </thead> <tbody> <tr> <td>1</td><td>One data centre with virtual machines for UB1</td><td rowspan="3" style="text-align: center;">Nearest Data Centre</td><td rowspan="3" style="text-align: center;">Round Robin</td></tr> <tr> <td>2</td><td>Two data centers with 25 and 50 Virtual Machines respectively for UB1</td><td></td></tr> <tr> <td>3</td><td>Three data centers with 100,75 and 25 Virtual Machines Respectively for UB1</td><td></td></tr> </tbody> </table> <p>i) Tabulate the overall response time and data processing of all the scenarios and plot the bar graph ii) Plot a line graph of data centre request servicing time of all the data centers for all the scenarios iii) Compare average response time by regions of all scenarios by plotting line graph iv) Mention the data centers used by the UB2, UB3, UB4 and UB5</p>	Scenario	Scenario Description	Load balancing algorithm	Service Broker Policy	1	One data centre with virtual machines for UB1	Nearest Data Centre	Round Robin	2	Two data centers with 25 and 50 Virtual Machines respectively for UB1		3	Three data centers with 100,75 and 25 Virtual Machines Respectively for UB1																													
Scenario	Scenario Description	Load balancing algorithm	Service Broker Policy																																								
1	One data centre with virtual machines for UB1	Nearest Data Centre	Round Robin																																								
2	Two data centers with 25 and 50 Virtual Machines respectively for UB1																																										
3	Three data centers with 100,75 and 25 Virtual Machines Respectively for UB1																																										
<p>3</p>	<p>Simulate the following scenarios for given data centre, data centre and virtual machine configuration and answer the following questions.</p> <p>Scenario 1: Closest data centre and round robin policies</p> <p>Scenario 2: Optimize response time and round robin policies Use the following userbase configuration for all the scenarios</p>																																										

	User Base	Region	Data Center	Peak Hour Users	Off Peak Hour Users	Virtual Machines
UB1	North America		DC1, DC3	1000	500	DC1-50 DC3-100
UB2	South America	--		1200	800	
UB3	Europe	DC4		2000	1000	DC4-150
UB4	Africa	--		500	300	

i) Tabulate and compare the Average response time and data processing time of all the scenarios by plotting the line graph

ii) Tabulate the response time of user bases in all scenarios and compare these by plotting bar graph. Which user base is taking maximum time among three scenarios? Why?

iii) Calculate the data transmission time from DC1 to UB2

iv) Plot the bar graph for data centre cost of all scenarios

4 Analyse the various service broker policies for the following configuration and answer the following questions.

Parameter	Value Used
UB Name	UB1
Region	2
Request Per User Per Hour	60
Data Size Per Request	100
Peak Hour Start (GMT)	3
Peak Hour End (GMT)	9
Avg Peak Users	40000
Avg Off Peak Users	4000
DC1 – No of VM	75
DC2 – No of VM	50
DC3 – No of VM	25
VM Image Size	10000 MB
VM Memory	512 MB
VM Bandwidth	1000 bps
DC1 – No of Physical Machine	2
DC2 – No of Physical Machine	2
DC3 – No of Physical Machine	2
DC – Memory Per Machine	204800 Mb
DC – Storage Per Machine	100000000 Mb
DC – Available BW Per Machine	1000000
DC – No of Processors Per Machine	4
DC – Processor Speed	10000 MIPS
DC – VM Policy	Time Shared
User Grouping Factor	1000
Request Grouping Factor	100
Executable Instruction Length	500
Load Balancing Policy	Throttled

	<p>a) Tabulate and compare the data processing time of service broker policies by plotting the line graph</p> <p>b) Tabulate and compare response time of service broker policies by plotting the bar graph</p> <p>c) Tabulate the cost for service broker policies and represent it using pie chart</p> <p>d) Which service broker policy is best and why?</p>																		
5	<p>Analyse the various load balancing algorithms for the given userbase, data centre and virtual machine configuration and answer the following questions. Consider the following userbase configuration for all load balancing algorithms.</p> <table border="1"> <tbody> <tr> <td>Number of User Bases</td><td>06</td></tr> <tr> <td>Region for the User Bases</td><td>UB1-South America, UB2-Asia, UB3-North America, UB4-Europe, UB5-Africa, UB6-Ocenia</td></tr> <tr> <td>Average Peak Users for All the User Bases</td><td>10000</td></tr> <tr> <td>Average off Peak Users for all the user bases</td><td>100</td></tr> <tr> <td>Peak Hours' Time</td><td>Depends on the region</td></tr> <tr> <td>Data centers in each user base</td><td>UB1-1, UB2-2, UB3-1, UB4-3, UB5-2, UB6-1</td></tr> <tr> <td>Virtual Machines in each data centre</td><td>6</td></tr> <tr> <td>Simulation time</td><td>10 mins</td></tr> <tr> <td>Service Broker Policy</td><td>Nearest Data Center</td></tr> </tbody> </table> <p>a) Tabulate and compare the data processing time of load balancing algorithms by plotting the line graph</p> <p>b) Tabulate the response time of load balancing algorithms by plotting the bar graph</p> <p>c) Tabulate the response time by region for load balancing algorithms and plot bar graph</p> <p>d) Which load balancing algorithm is best and why?</p>	Number of User Bases	06	Region for the User Bases	UB1-South America, UB2-Asia, UB3-North America, UB4-Europe, UB5-Africa, UB6-Ocenia	Average Peak Users for All the User Bases	10000	Average off Peak Users for all the user bases	100	Peak Hours' Time	Depends on the region	Data centers in each user base	UB1-1, UB2-2, UB3-1, UB4-3, UB5-2, UB6-1	Virtual Machines in each data centre	6	Simulation time	10 mins	Service Broker Policy	Nearest Data Center
Number of User Bases	06																		
Region for the User Bases	UB1-South America, UB2-Asia, UB3-North America, UB4-Europe, UB5-Africa, UB6-Ocenia																		
Average Peak Users for All the User Bases	10000																		
Average off Peak Users for all the user bases	100																		
Peak Hours' Time	Depends on the region																		
Data centers in each user base	UB1-1, UB2-2, UB3-1, UB4-3, UB5-2, UB6-1																		
Virtual Machines in each data centre	6																		
Simulation time	10 mins																		
Service Broker Policy	Nearest Data Center																		

Results of the Simulation Completed at: 08/12/2021 11:29:47

Overall Response Time Summary

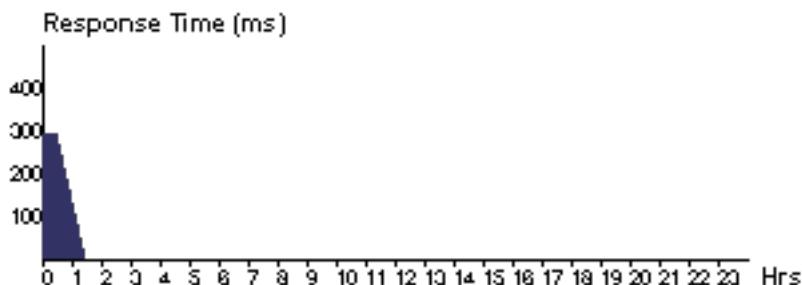
	Avg (ms)	Min (ms)	Max (ms)
Overall response time:	275.95	38.16	642.70
Data Center processing time:	0.41	0.02	1.04

Response Time by Region

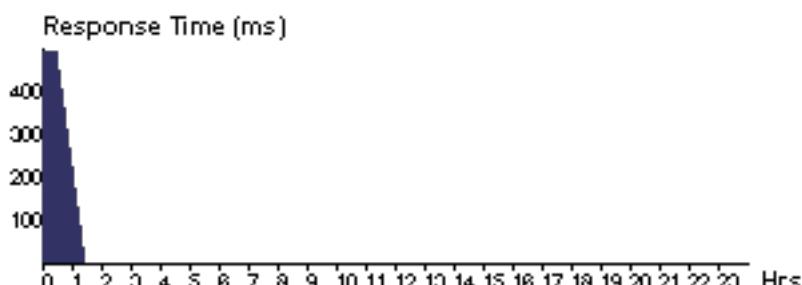
Userbase	Avg (ms)	Min (ms)	Max (ms)
UB1	299.89	229.71	376.68
UB2	499.23	375.18	642.70
UB3	50.17	38.16	63.18
UB4	300.05	235.63	381.14
UB5	300.01	235.64	370.65
UB6	401.52	312.12	506.12

User Base Hourly Response Times

UB1

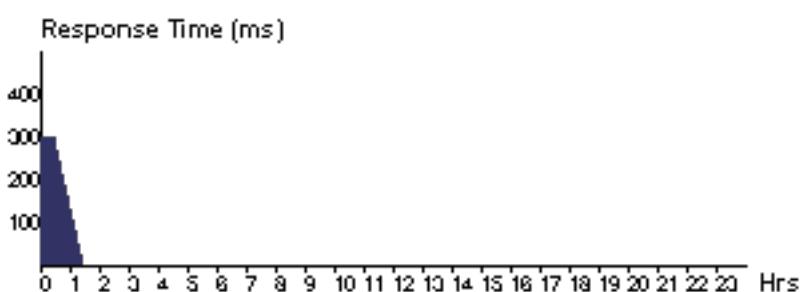
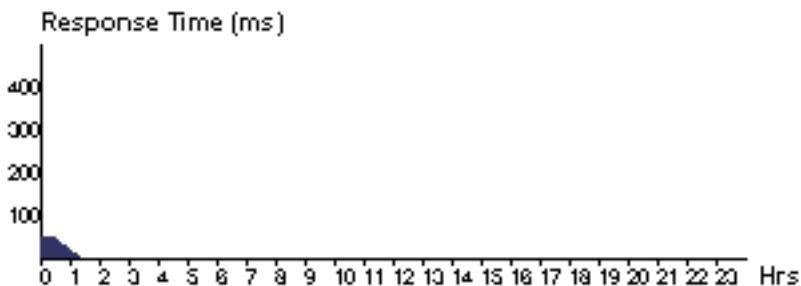


UB2

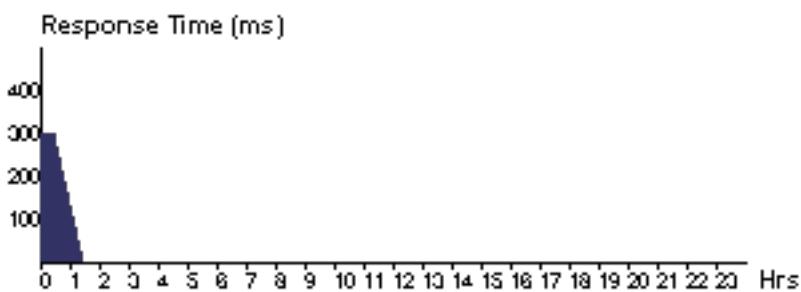


UB3

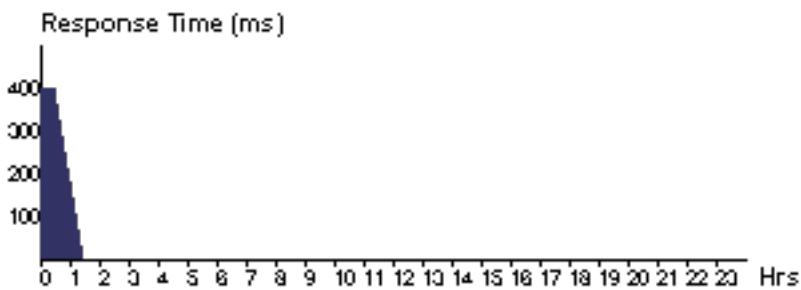
UB4



UB5



UB6



Data Center Request Servicing Times

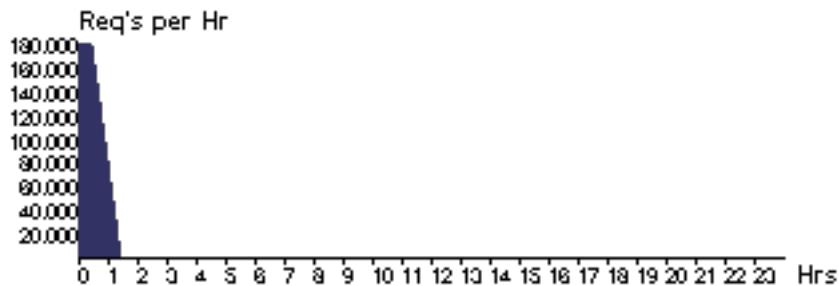
Data Center	Avg (ms)	Min (ms)	Max (ms)
DC1	0.41	0.02	1.04

Data Center Hourly Average Processing Times

DC1

Data Center Hourly Loading

DC1



Cost

Total Virtual Machine Cost (\$): 0.50

Total Data Transfer Cost (\$): 1.91

Grand Total: (\$) 2.41

Data Center	VM Cost \$	Data Transfer Cost \$	Total \$
DC1	0.50	1.91	2.41

Results of the Simulation Completed at: 08/12/2021 11:33:16

Overall Response Time Summary

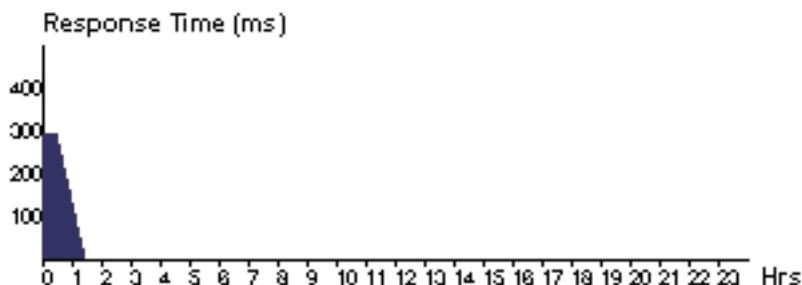
	Avg (ms)	Min (ms)	Max (ms)
Overall response time:	275.87	38.91	642.73
Data Center processing time:	0.41	0.02	1.00

Response Time by Region

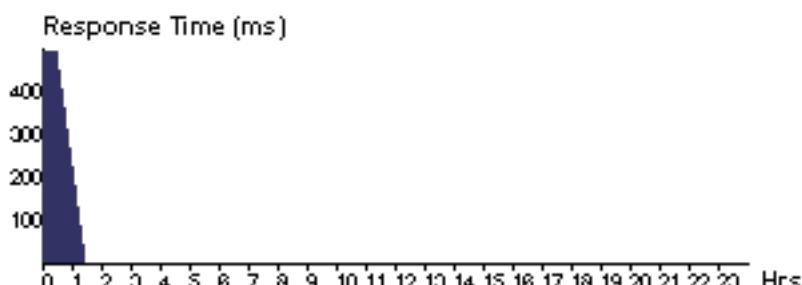
Userbase	Avg (ms)	Min (ms)	Max (ms)
UB1	299.76	228.17	376.66
UB2	498.81	370.18	642.73
UB3	50.23	38.91	62.92
UB4	300.56	234.13	378.15
UB5	300.16	225.13	384.14
UB6	400.77	322.11	490.12

User Base Hourly Response Times

UB1

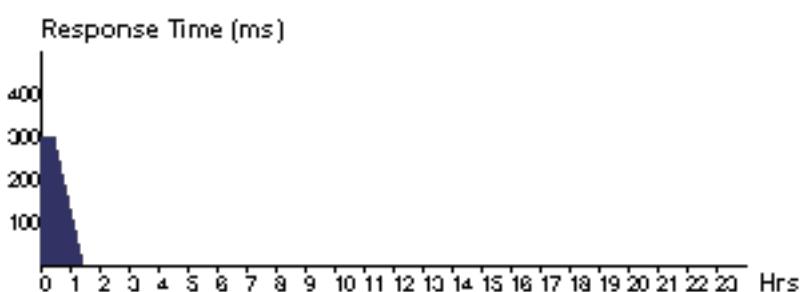
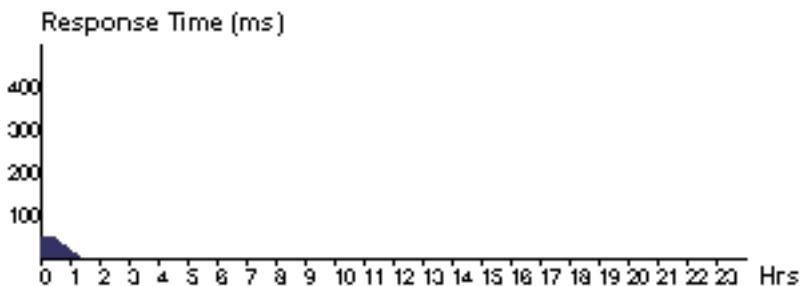


UB2

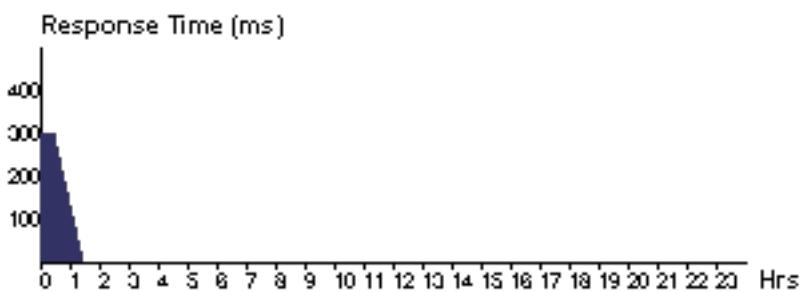


UB3

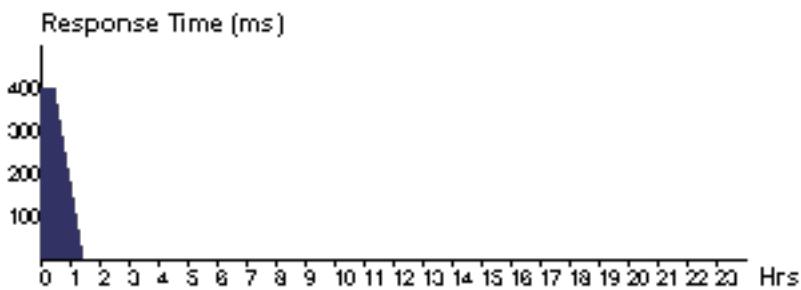
UB4



UB5



UB6



Data Center Request Servicing Times

Data Center	Avg (ms)	Min (ms)	Max (ms)
DC1	0.41	0.02	1.00

Data Center Hourly Average Processing Times

DC1

Data Center Hourly Loading

DC1



Cost

Total Virtual Machine Cost (\$): 0.50

Total Data Transfer Cost (\$): 1.91

Grand Total: (\$) 2.41

Data Center	VM Cost \$	Data Transfer Cost \$	Total \$
DC1	0.50	1.91	2.41

Results of the Simulation Completed at: 08/12/2021 11:38:40

Overall Response Time Summary

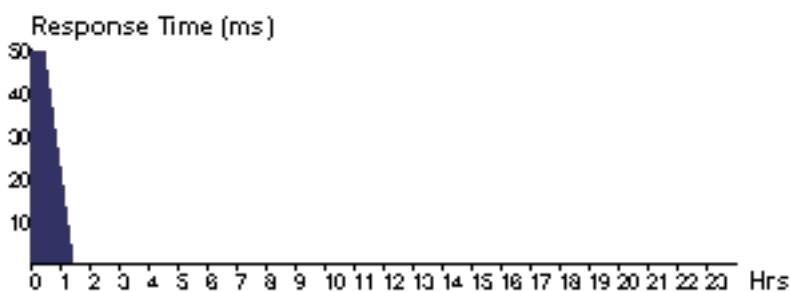
	Avg (ms)	Min (ms)	Max (ms)
Overall response time:	50.09	39.55	61.61
Data Center processing time:	0.48	0.01	0.85

Response Time by Region

Userbase	Avg (ms)	Min (ms)	Max (ms)
UB1	50.09	39.55	61.61

User Base Hourly Response Times

UB1



Data Center Request Servicing Times

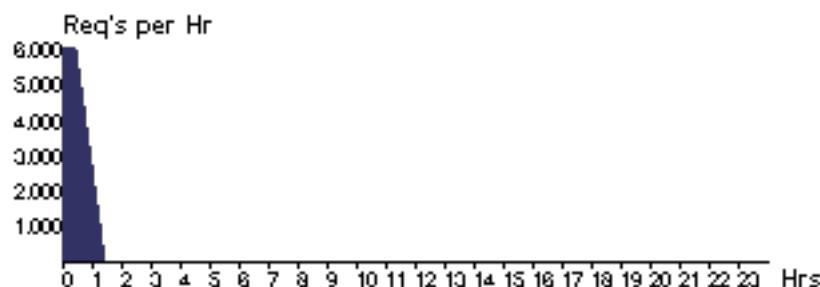
Data Center	Avg (ms)	Min (ms)	Max (ms)
DC1	0.48	0.01	0.85

Data Center Hourly Average Processing Times

DC1

Data Center Hourly Loading

DC1



Cost

Total Virtual Machine Cost (\$): 0.51

Total Data Transfer Cost (\$): 0.06

Grand Total: (\$) 0.57

Data Center	VM Cost \$	Data Transfer Cost \$	Total \$
DC1	0.51	0.06	0.57

Results of the Simulation Completed at: 08/12/2021 11:45:12

Overall Response Time Summary

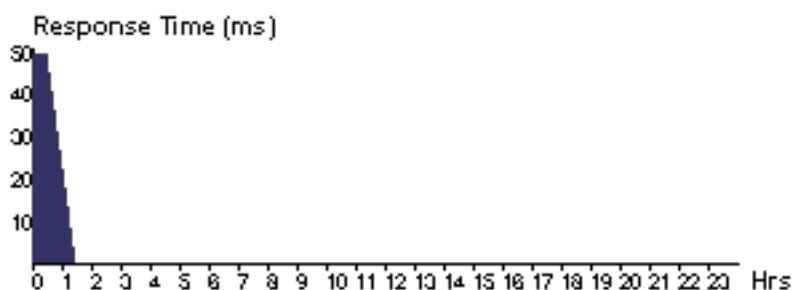
	Avg (ms)	Min (ms)	Max (ms)
Overall response time:	50.77	40.01	61.83
Data Center processing time:	1.16	0.07	2.00

Response Time by Region

Userbase	Avg (ms)	Min (ms)	Max (ms)
UB1	50.77	40.01	61.83

User Base Hourly Response Times

UB1



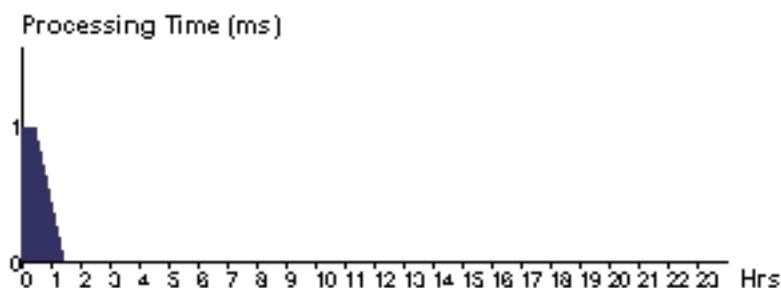
Data Center Request Servicing Times

Data Center	Avg (ms)	Min (ms)	Max (ms)
DC1	0.67	0.07	1.08
DC2	1.59	0.13	2.00

Data Center Hourly Average Processing Times

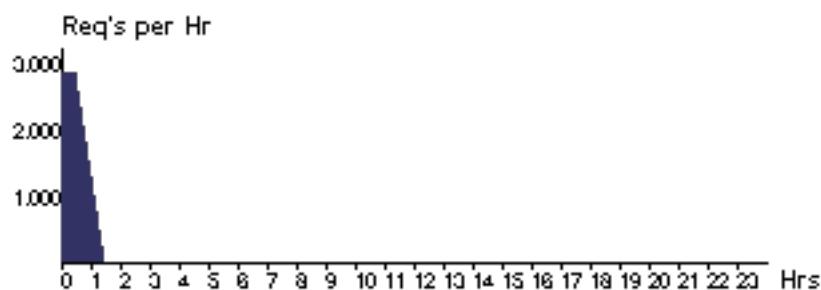
DC1

DC2

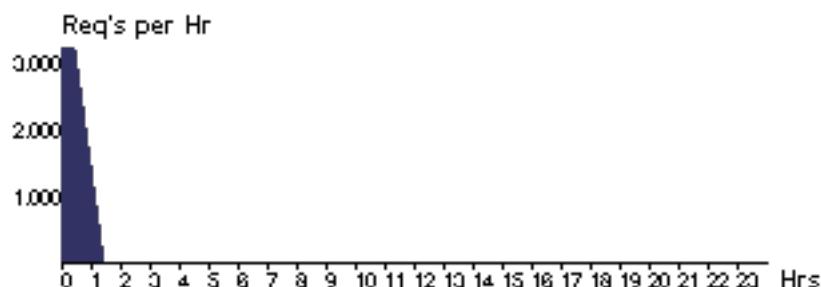


Data Center Hourly Loading

DC1



DC2



Cost

Total Virtual Machine Cost (\$): 7.61

Total Data Transfer Cost (\$): 0.06

Grand Total: (\$) 7.67

Data Center	VM Cost \$	Data Transfer Cost \$	Total \$
DC2	5.07	0.03	5.10
DC1	2.54	0.03	2.57

Results of the Simulation Completed at: 08/12/2021 11:49:26

Overall Response Time Summary

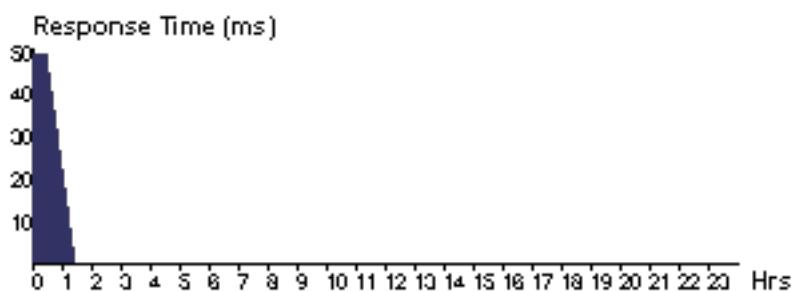
	Avg (ms)	Min (ms)	Max (ms)
Overall response time:	50.70	39.56	62.38
Data Center processing time:	1.09	0.05	2.63

Response Time by Region

Userbase	Avg (ms)	Min (ms)	Max (ms)
UB1	50.70	39.56	62.38

User Base Hourly Response Times

UB1



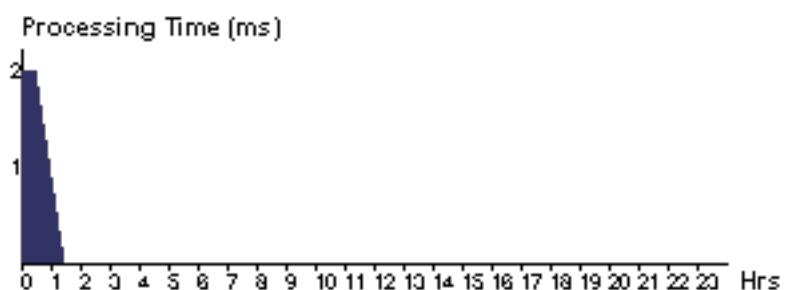
Data Center Request Servicing Times

Data Center	Avg (ms)	Min (ms)	Max (ms)
DC1	0.70	0.07	1.08
DC2	2.20	0.19	2.63
DC3	0.48	0.05	0.88

Data Center Hourly Average Processing Times

DC1

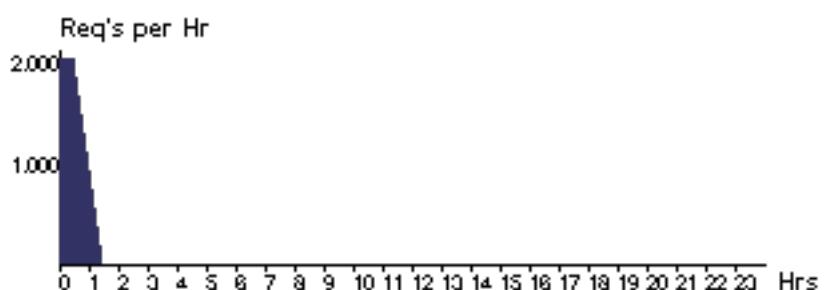
DC2



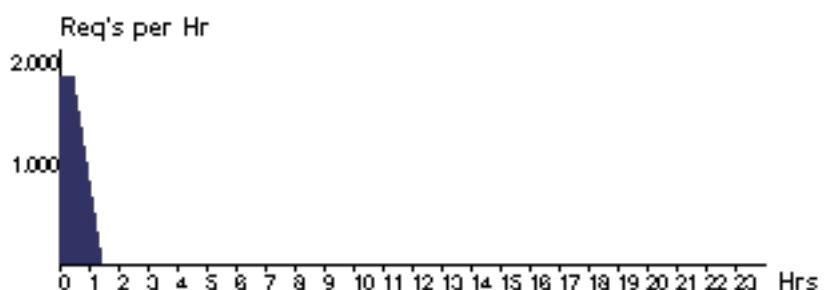
DC3

Data Center Hourly Loading

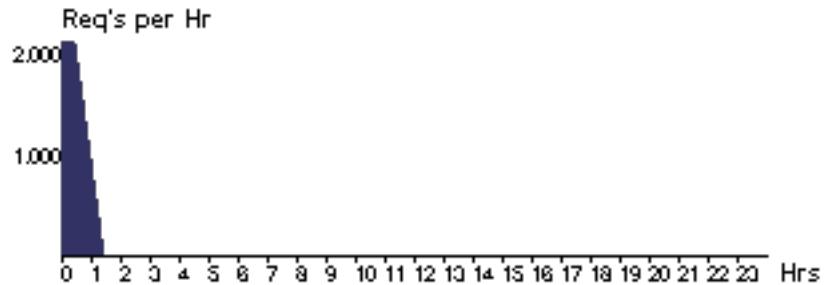
DC1



DC2



DC3



Cost

Total Virtual Machine Cost (\$): 10.65

Total Data Transfer Cost (\$): 0.06

Grand Total: (\$) 10.71

Data Center	VM Cost \$	Data Transfer Cost \$	Total \$
DC2	7.61	0.02	7.62
DC1	2.54	0.02	2.56
DC3	0.51	0.02	0.53

Results of the Simulation Completed at: 08/12/2021 12:00:27

Overall Response Time Summary

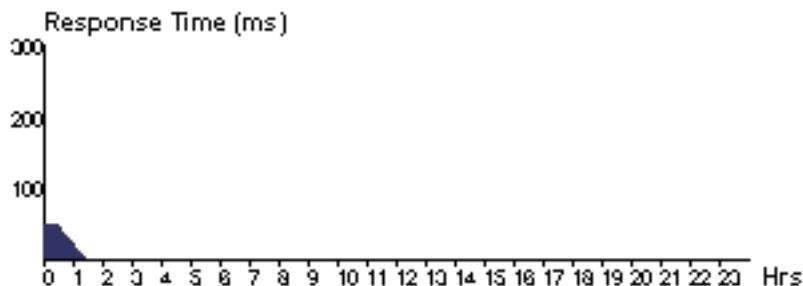
	Avg (ms)	Min (ms)	Max (ms)
Overall response time:	113.93	38.42	365.26
Data Center processing time:	2.80	0.19	4.55

Response Time by Region

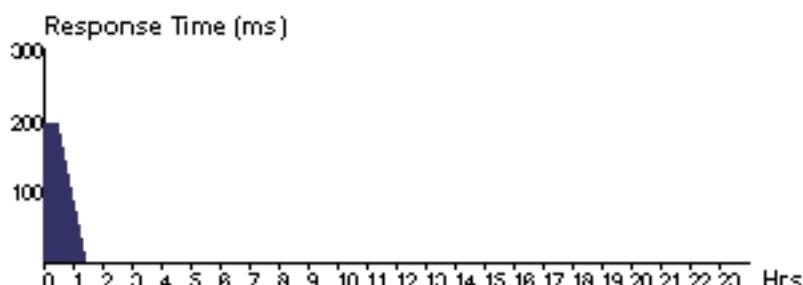
Userbase	Avg (ms)	Min (ms)	Max (ms)
UB1	51.63	38.42	65.29
UB2	201.31	151.64	256.70
UB3	53.76	41.53	65.79
UB4	303.71	233.26	365.26

User Base Hourly Response Times

UB1

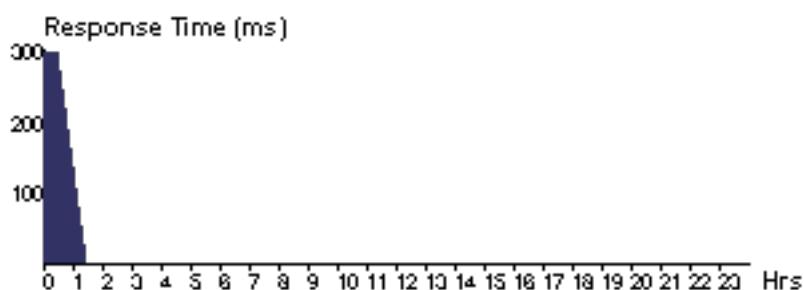
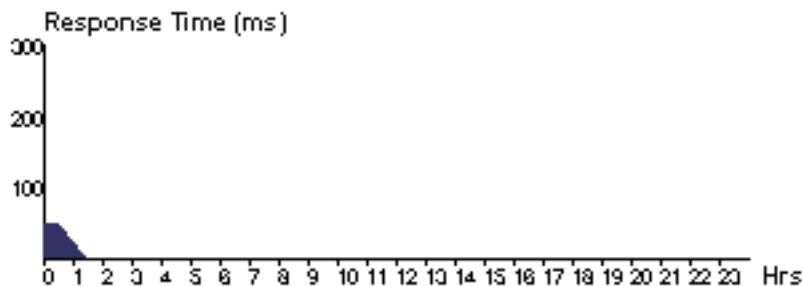


UB2



UB3

UB4



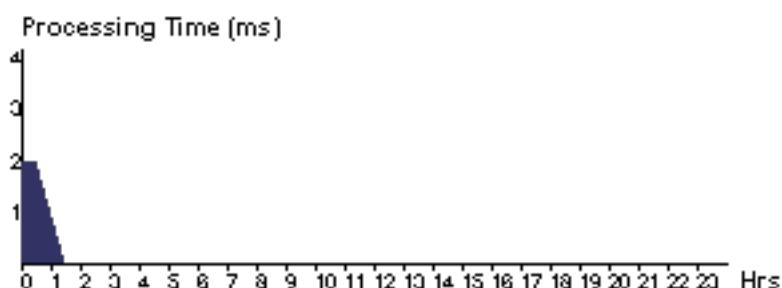
Data Center Request Servicing Times

Data Center	Avg (ms)	Min (ms)	Max (ms)
DC1	0.79	0.19	1.41
DC3	2.66	0.57	3.28
DC4	4.11	0.38	4.55

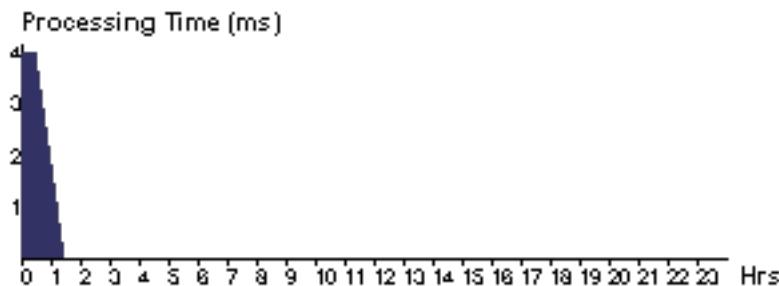
Data Center Hourly Average Processing Times

DC1

DC3



DC4



Data Center Hourly Loading

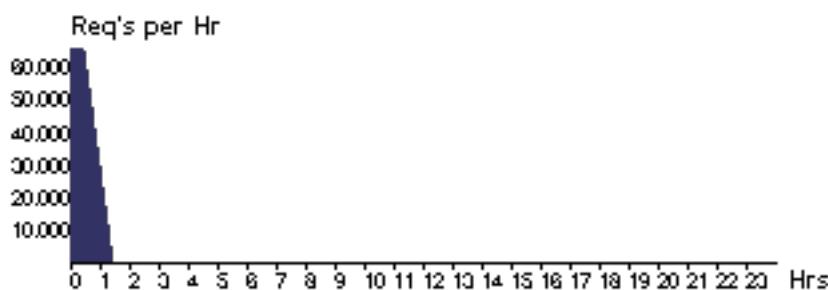
DC1



DC3



DC4



Cost

Total Virtual Machine Cost (\$): 30.11

Total Data Transfer Cost (\$): 1.52

Grand Total: (\$) 31.64

Data Center	VM Cost \$	Data Transfer Cost \$	Total \$
DC1	5.02	0.42	5.44

DC4	15.06	0.69	15.74
DC3	10.04	0.42	10.45

Results of the Simulation Completed at: 08/12/2021 12:01:51

Overall Response Time Summary

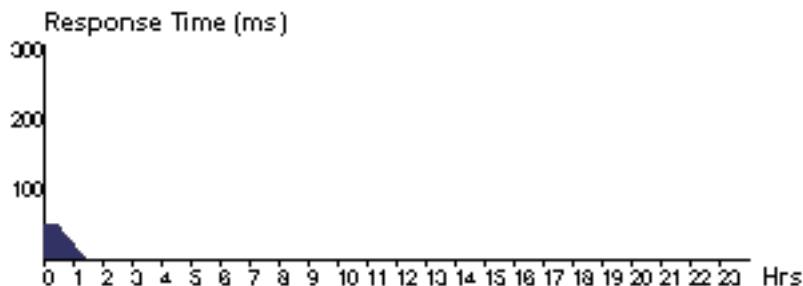
	Avg (ms)	Min (ms)	Max (ms)
Overall response time:	114.01	38.40	392.26
Data Center processing time:	2.75	0.19	4.56

Response Time by Region

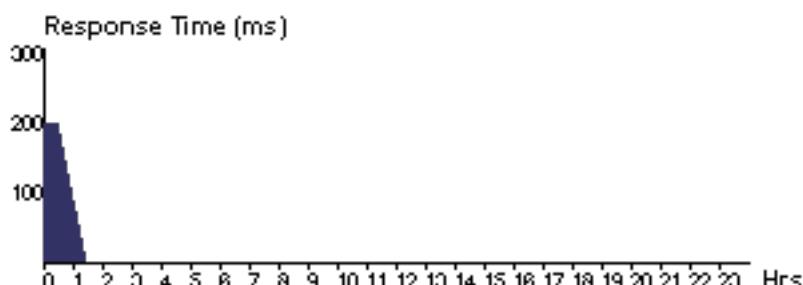
Userbase	Avg (ms)	Min (ms)	Max (ms)
UB1	51.41	38.40	64.28
UB2	201.59	152.73	253.58
UB3	53.78	40.81	66.56
UB4	304.31	242.27	392.26

User Base Hourly Response Times

UB1

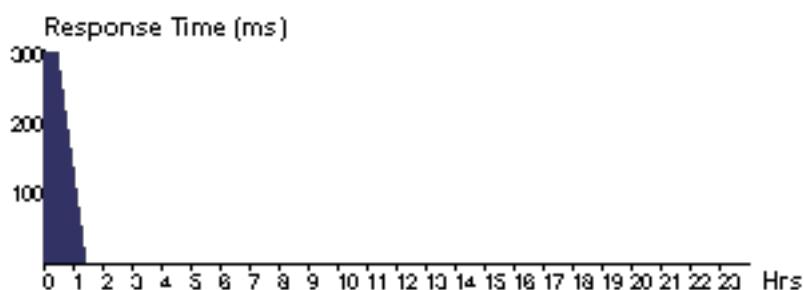
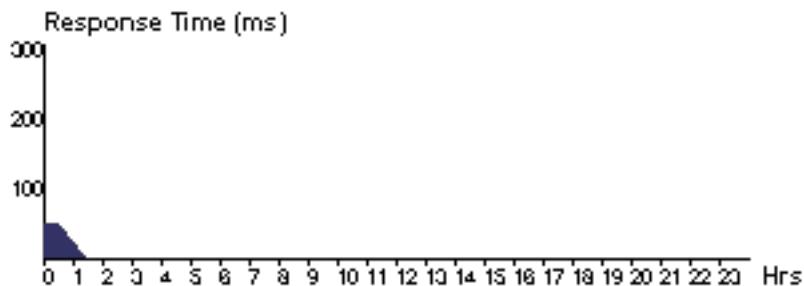


UB2



UB3

UB4



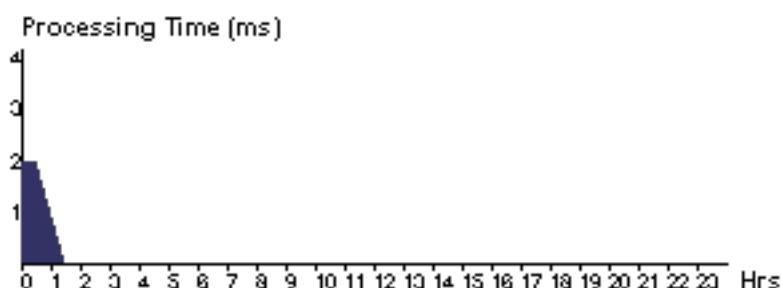
Data Center Request Servicing Times

Data Center	Avg (ms)	Min (ms)	Max (ms)
DC1	0.81	0.19	1.41
DC3	2.65	0.57	3.28
DC4	4.11	0.38	4.56

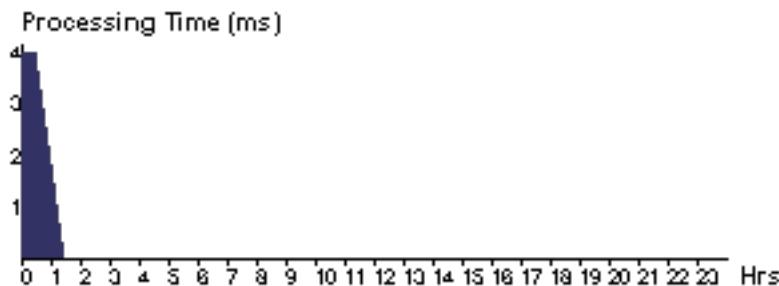
Data Center Hourly Average Processing Times

DC1

DC3



DC4



Data Center Hourly Loading

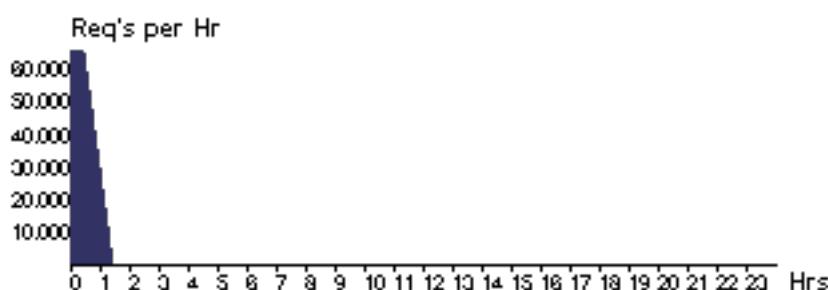
DC1



DC3



DC4



Cost

Total Virtual Machine Cost (\$):	30.11
Total Data Transfer Cost (\$):	1.52
Grand Total: (\$)	31.64

Data Center	VM Cost \$	Data Transfer Cost \$	Total \$
DC1	5.02	0.46	5.48

DC4	15.06	0.69	15.74
DC3	10.04	0.37	10.41

Results of the Simulation Completed at: 08/12/2021 12:21:46

Overall Response Time Summary

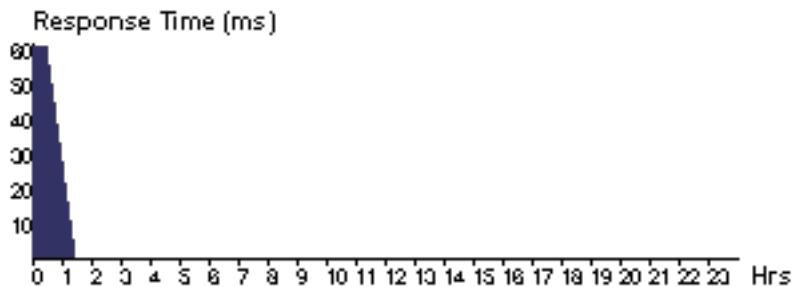
	Avg (ms)	Min (ms)	Max (ms)
Overall response time:	62.14	44.82	100.78
Data Center processing time:	12.62	0.32	47.53

Response Time by Region

Userbase	Avg (ms)	Min (ms)	Max (ms)
UB1	62.14	44.82	100.78

User Base Hourly Response Times

UB1



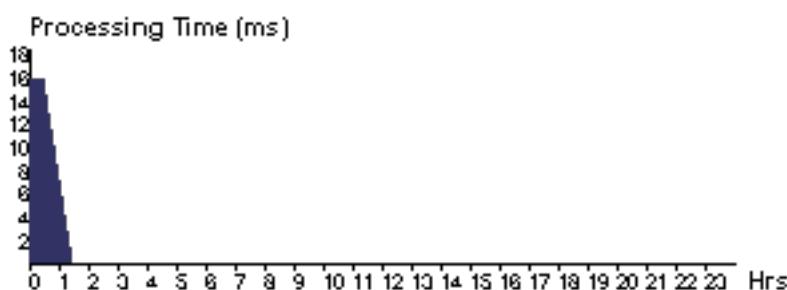
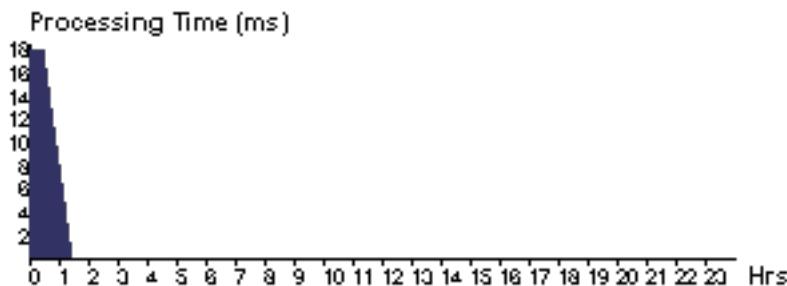
Data Center Request Servicing Times

Data Center	Avg (ms)	Min (ms)	Max (ms)
DC1	18.36	0.72	47.53
DC2	16.42	0.32	31.19
DC3	3.72	0.45	5.75

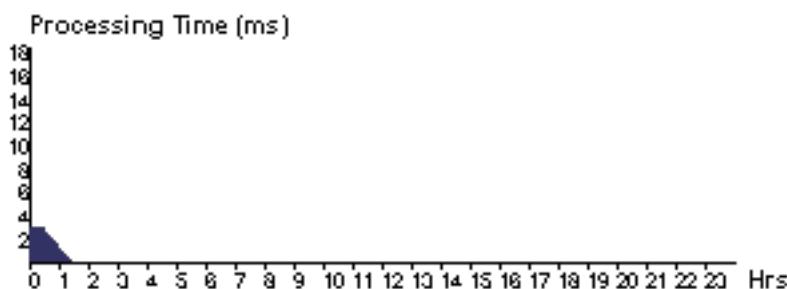
Data Center Hourly Average Processing Times

DC1

DC2

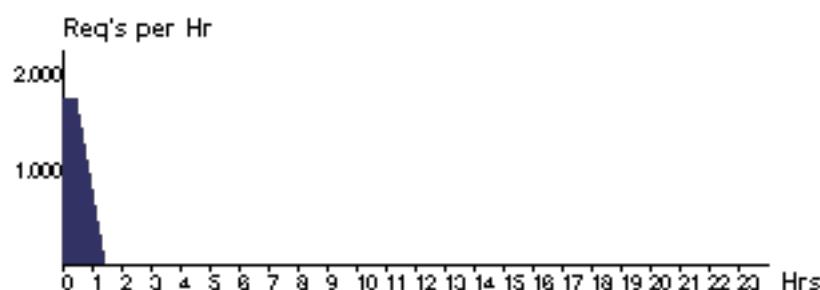


DC3

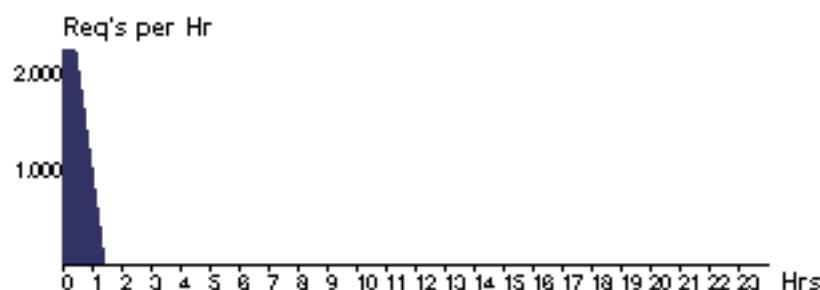


Data Center Hourly Loading

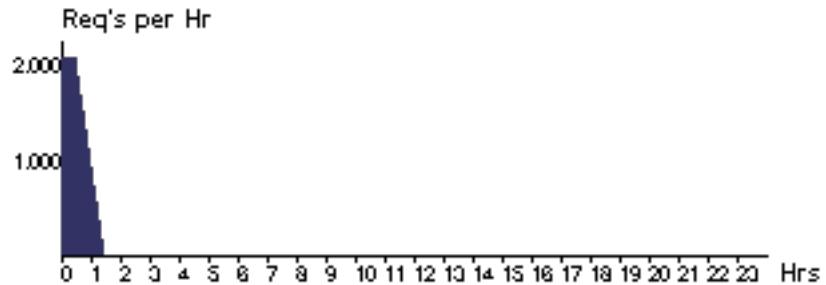
DC1



DC2



DC3



Cost

Total Virtual Machine Cost (\$): 13.18

Total Data Transfer Cost (\$): 0.06

Grand Total: (\$) 13.24

Data Center	VM Cost \$	Data Transfer Cost \$	Total \$
DC2	5.07	0.02	5.09
DC1	7.61	0.02	7.62
DC3	0.51	0.02	0.53

Results of the Simulation Completed at: 08/12/2021 12:14:48

Overall Response Time Summary

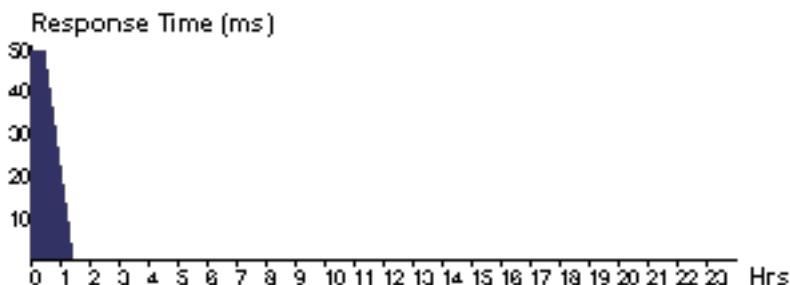
	Avg (ms)	Min (ms)	Max (ms)
Overall response time:	50.16	37.62	60.91
Data Center processing time:	0.50	0.02	0.92

Response Time by Region

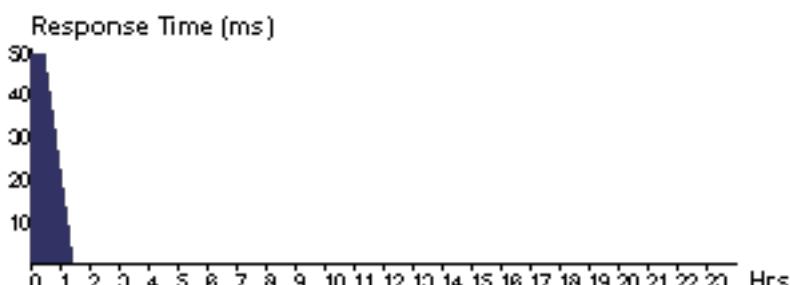
Userbase	Avg (ms)	Min (ms)	Max (ms)
UB1	50.12	37.62	60.37
UB2	50.46	42.17	58.68
UB3	50.18	42.16	60.91
UB4	49.25	39.16	56.66
UB5	50.69	42.66	57.66
UB6	50.31	42.61	59.91

User Base Hourly Response Times

UB1

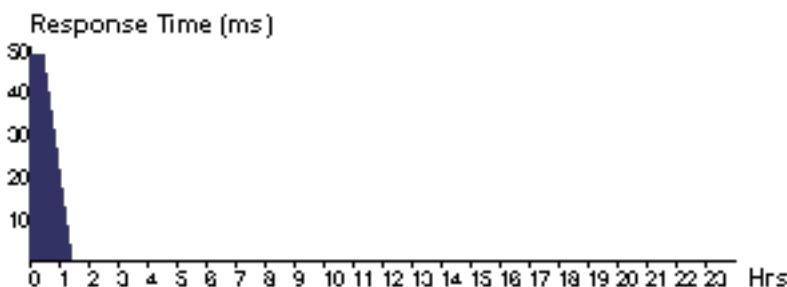
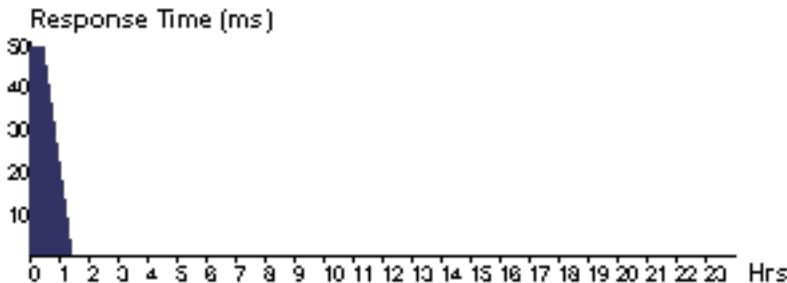


UB2

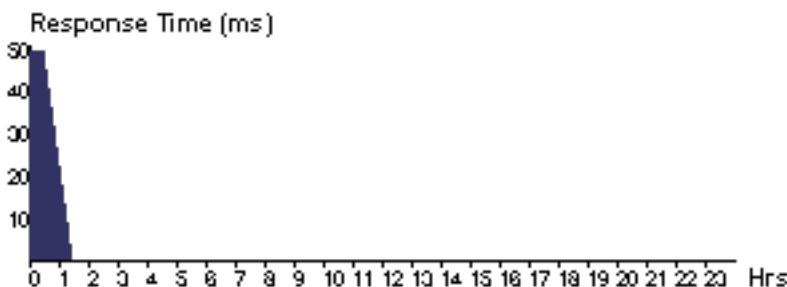


UB3

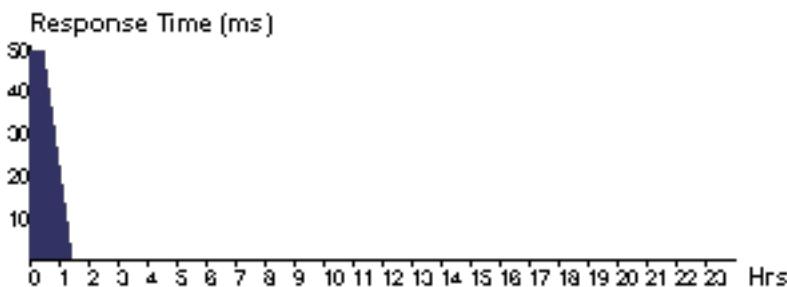
UB4



UB5



UB6



Data Center Request Servicing Times

Data Center	Avg (ms)	Min (ms)	Max (ms)
DC10	0.51	0.11	0.90
DC1	0.40	0.03	0.86
DC2	0.54	0.12	0.92
DC3	0.51	0.10	0.92
DC4	0.52	0.02	0.91
DC5	0.47	0.08	0.90
DC6	0.53	0.09	0.90
DC7	0.49	0.15	0.90
DC8	0.50	0.14	0.91

DC9	0.57	0.02	0.91
-----	------	------	------

Data Center Hourly Average Processing Times

DC1

DC10

DC2

DC3

DC4

DC5

DC6

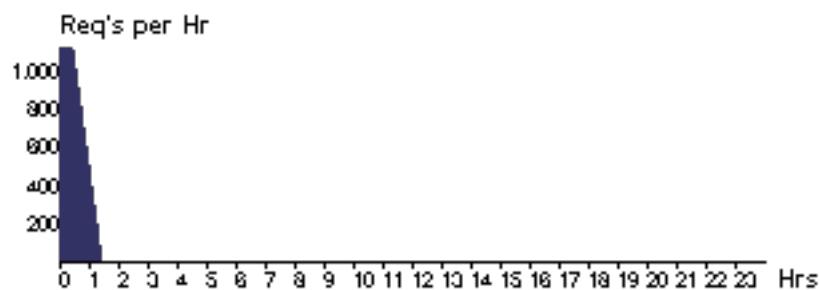
DC7

DC8

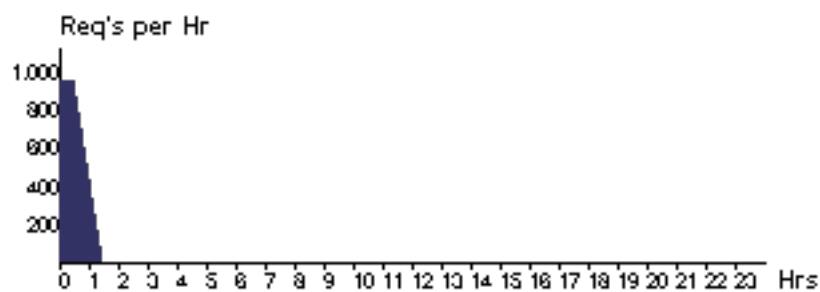
DC9

Data Center Hourly Loading

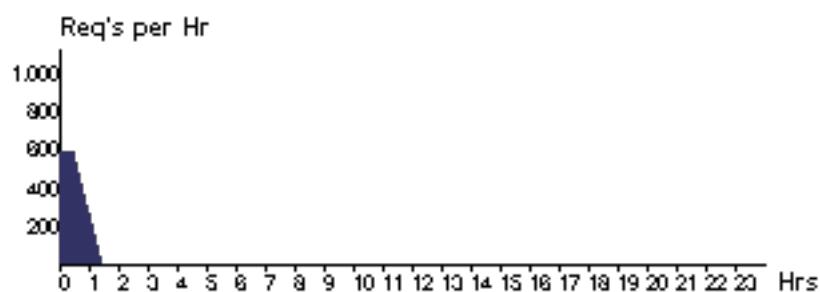
DC1



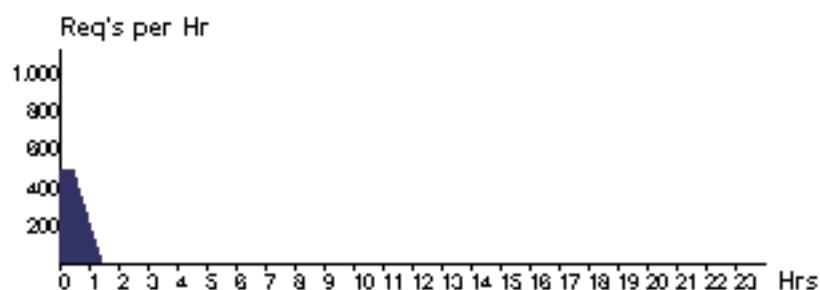
DC10



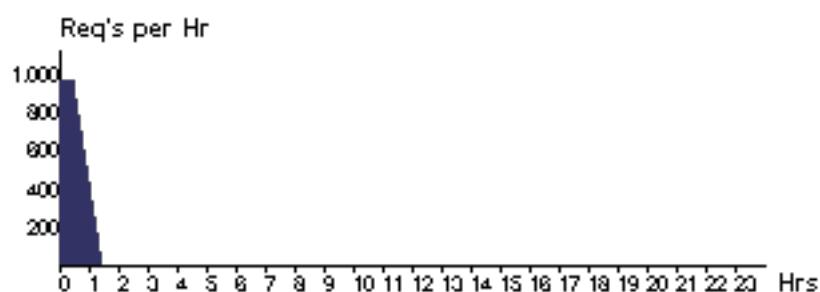
DC2



DC3

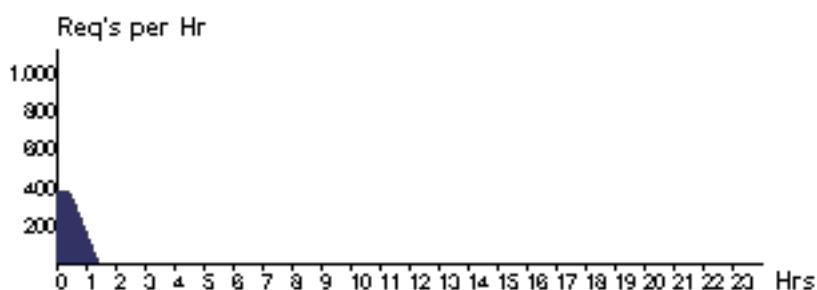
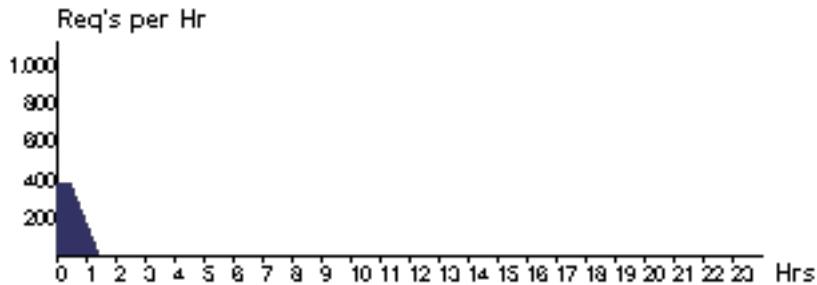


DC4

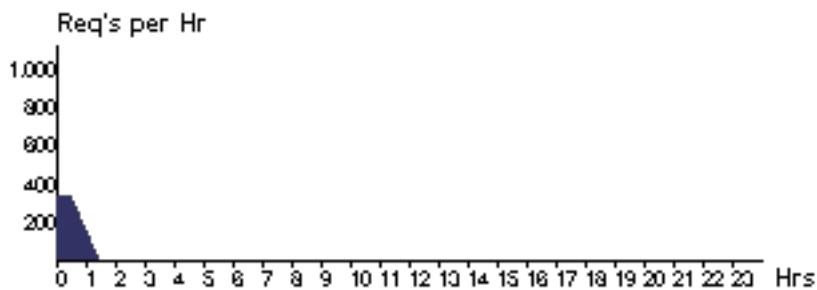


DC5

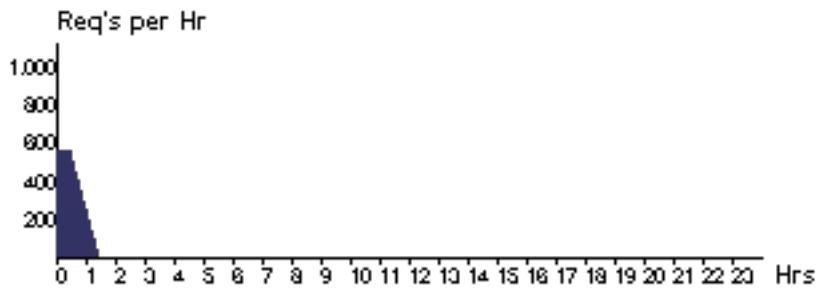
DC6



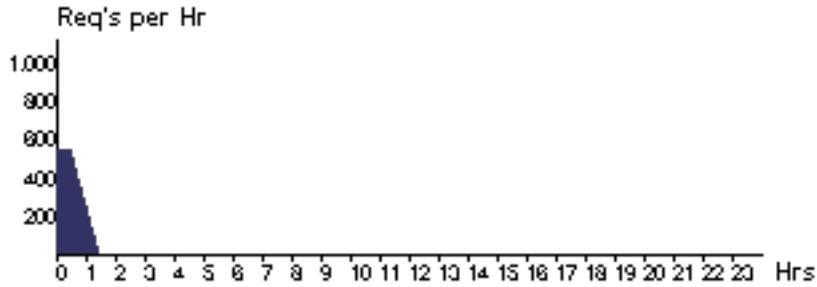
DC7



DC8



DC9



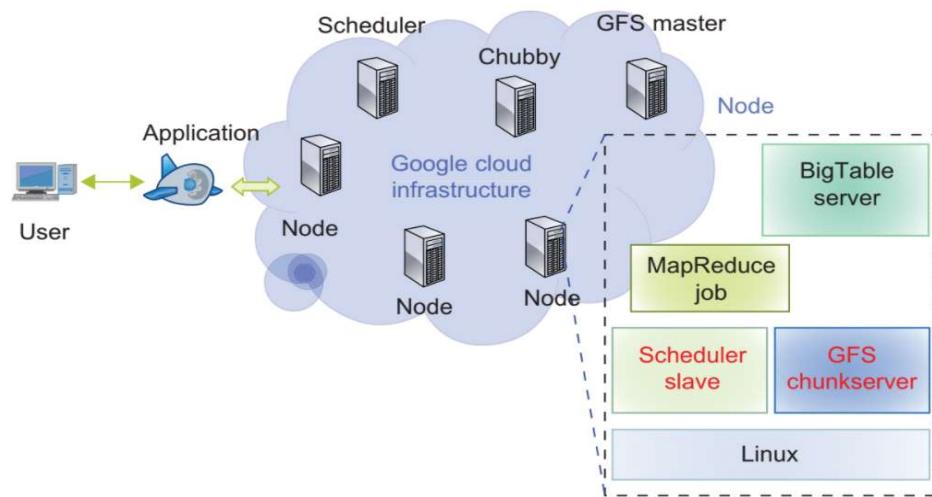
Cost

Total Virtual Machine Cost (\$):	0.99
Total Data Transfer Cost (\$):	0.07
Grand Total: (\$)	1.05

Data Center	VM Cost \$	Data Transfer Cost \$	Total \$
DC10	0.10	0.01	0.11
DC2	0.10	0.01	0.11
DC1	0.08	0.01	0.10
DC4	0.10	0.01	0.11
DC3	0.10	0.01	0.11
DC6	0.10	0.00	0.10
DC5	0.10	0.00	0.10
DC8	0.10	0.01	0.11
DC7	0.10	0.00	0.10
DC9	0.10	0.01	0.11

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. 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.



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". 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. Java Server 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 datacentre 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.

Bulk downloading

SDK version 1.2.2 adds support for bulk downloads of data using Python. The open-source Python projects gaebar, appocket, and gawsh also allow users to download and back up App Engine data. No method for bulk downloading data from GAE using Java currently exists.

Restrictions

- Developers have read-only access to the filesystem on App Engine. Applications can use only virtual filesystems, like gae-filestore.
- App Engine can only execute code called from an HTTP request (scheduled background tasks allow for self-calling HTTP requests).
- Users may upload arbitrary Python modules, but only if they are pure-Python; C and Pyrex modules are not supported.
- Java applications may only use a subset (The JRE Class White List) of the classes from the JRE standard edition. This restriction does not exist with the App Engine Standard Java8 runtime.
- A process started on the server to answer a request can't last more than 60 seconds (with the 1.4.0 release, this restriction does not apply to background jobs anymore).
- Does not support sticky sessions (a.k.a. session affinity), only replicated sessions are supported including limitation of the amount of data being serialized and time for session serialization.

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.

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.

Google App Engine Features:

- Blob store for serving large data objects;
- GAE Cloud Storage for storing data objects;
- Page Speed Service for automatically speeding up webpage load times;
- URL Fetch Service to issue HTTP requests and receive responses for efficiency and scaling; and
- Memcache for a fully managed in-memory data store.

Benefits of GAE

- Ease of setup and use. GAE is fully managed, so users can write code without considering IT operations and back-end infrastructure. The built-in APIs enable users to build different types of applications. Access to application logs also facilitates debugging and monitoring in production.
- Pay-per-use pricing. GAE's billing scheme only charges users daily for the resources they use. Users can monitor their resource usage and bills on a dashboard.
- Scalability. Google App Engine automatically scales as workloads fluctuate, adding and removing application instances or application resources as needed.
- Security. GAE supports the ability to specify a range of acceptable Internet Protocol (IP) addresses. Users can allow list specific networks and services and blocklist specific IP addresses.

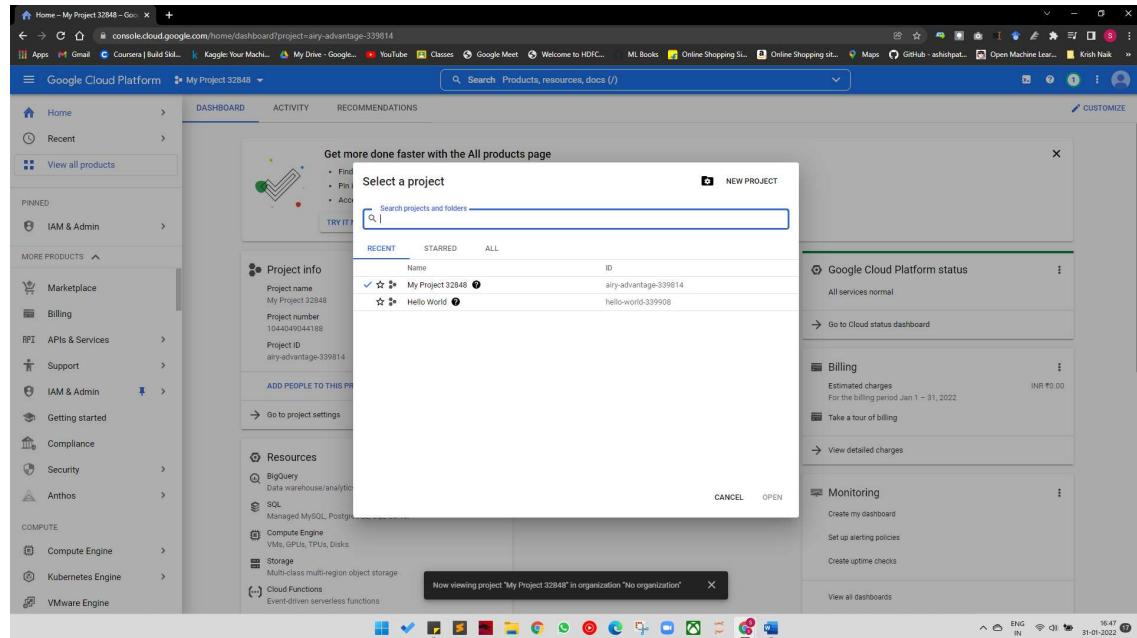
GAE challenges

- Lack of control. Although a managed infrastructure has advantages, if a problem occurs in the back-end infrastructure, the user is dependent on Google to fix it.
- Performance limits. CPU-intensive operations are slow and expensive to perform using GAE. This is because one physical server may be serving several separate, unrelated app engine users at once who need to share the CPU.
- Limited access. Developers have limited, read-only access to the GAE filesystem.
- Java limits. Java apps cannot create new threads and can only use a subset of the Java runtime environment standard edition classes.

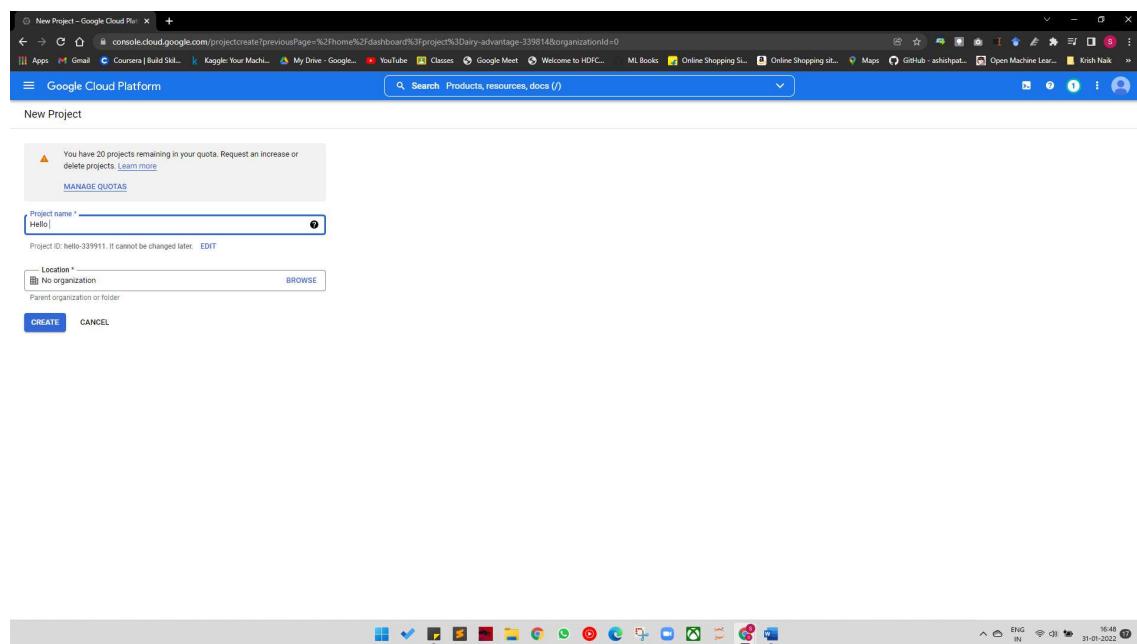
Install Google App Engine. Create hello world app and other simple web applications using python/java

Step 1: Create an account on Google App Engine.

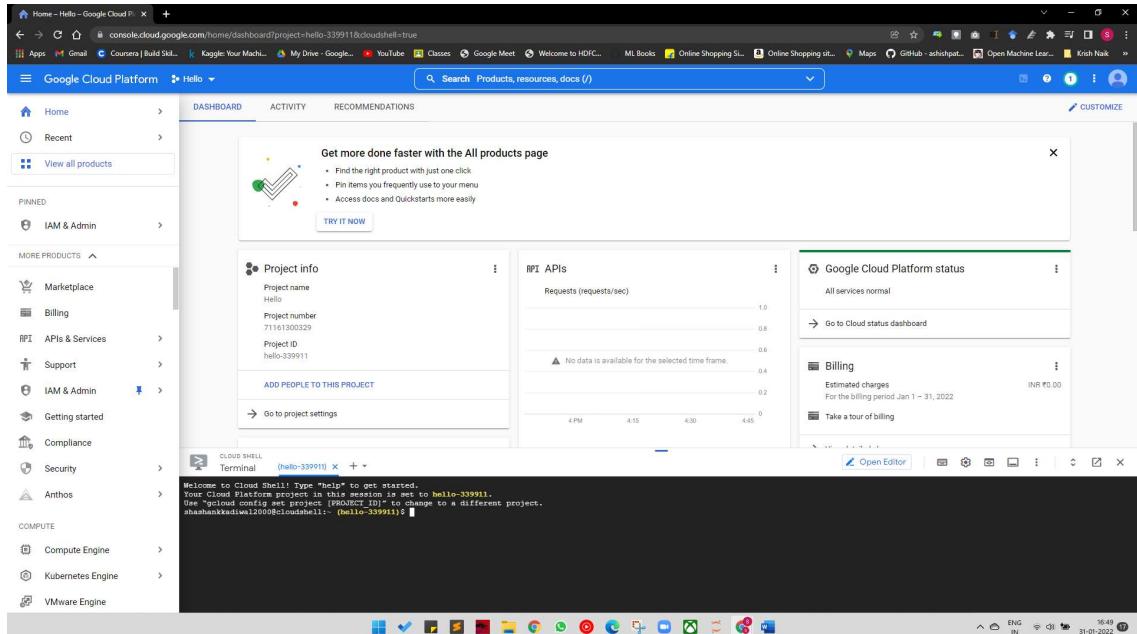
Step 2: Log in to your Google App Engine Account.



Step 3: Create a New Project and give a Project Name and click on create.



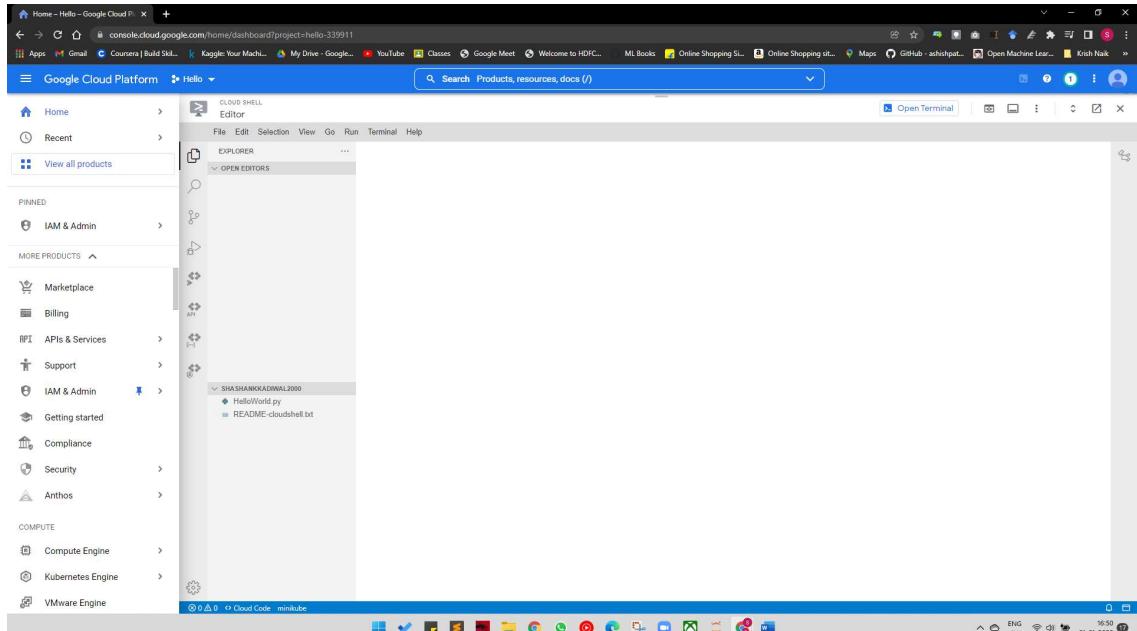
Step 4: Once the project is created click on **Activate Cloud Shell** on the top right corner.



Step 5: Once the cloud shell gets activated type the following command in the terminal and choose the region as South Asia

gcloud app create

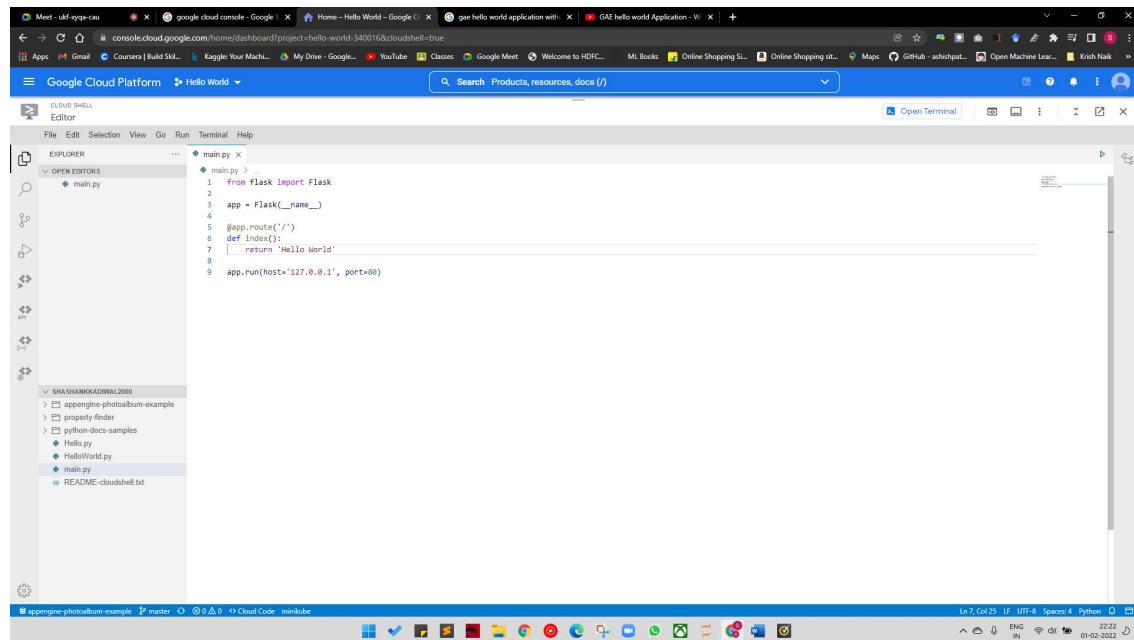
Click on Open Editor.



Step 6: Click on File → New File.

Step 7: Name the File as main.py and type the following code:

```
from flask import Flask  
  
app = Flask(__name__)  
  
@app.route('/')  
  
def index():  
  
    return 'Hello World'  
  
app.run(host='127.0.0.1', port=80)
```

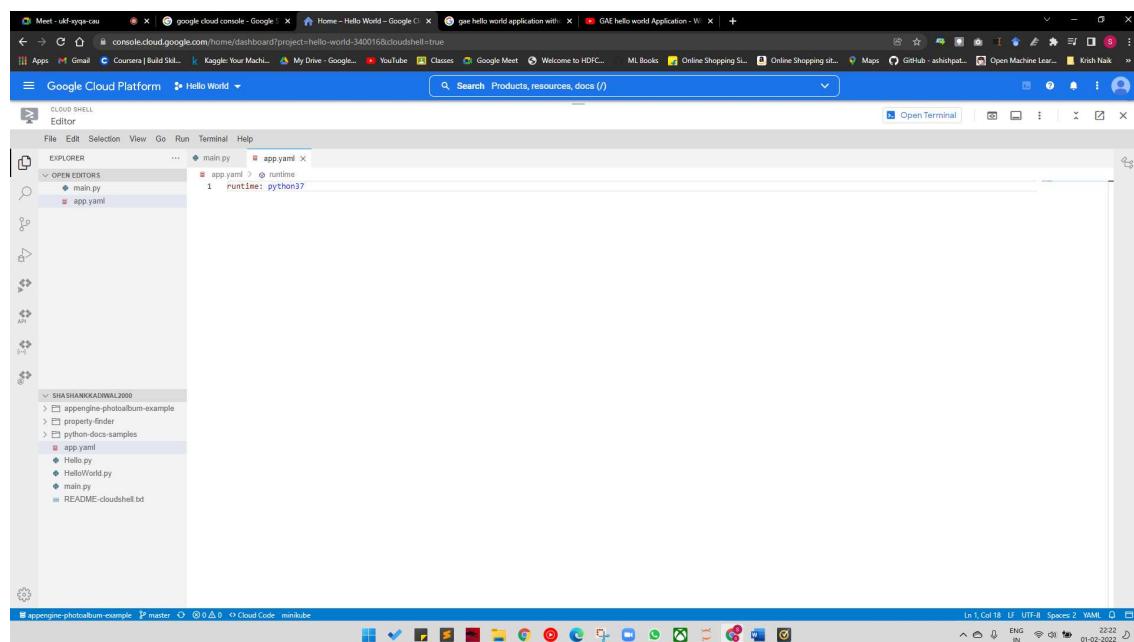


The screenshot shows the Google Cloud Platform Cloud Shell interface. In the center, there is an 'EXPLORER' panel showing a directory structure under 'SHASHANKKADWAL2000'. Inside the 'main' folder, there is a 'main.py' file. The code content is displayed in the main editor area:

```
1 from flask import Flask  
2  
3 app = Flask(__name__)  
4  
5 @app.route('/')  
6 def index():  
7     return 'Hello World'  
8  
9 app.run(host='127.0.0.1', port=80)
```

The status bar at the bottom indicates the file is in 'master' branch, has 0 changes, and is using Python 3.7.

Step8: Create a app.yaml file



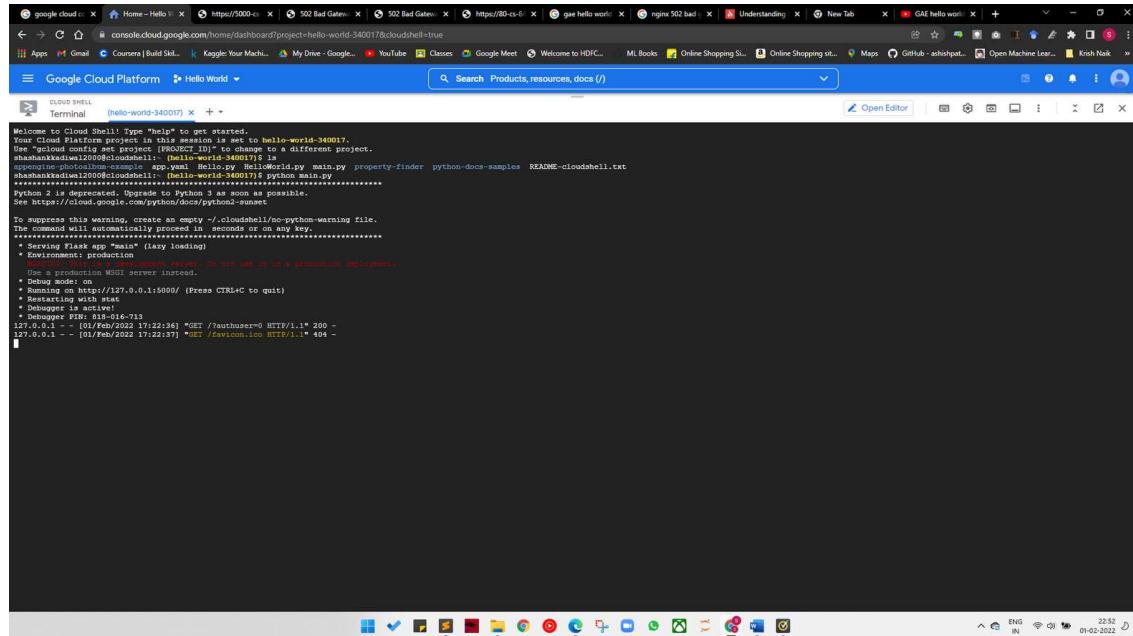
The screenshot shows the Google Cloud Platform Cloud Shell interface. In the center, there is an 'EXPLORER' panel showing a directory structure under 'SHASHANKKADWAL2000'. Inside the 'main' folder, there are now two files: 'main.py' and 'app.yaml'. The 'app.yaml' file contains the following configuration:

```
1 runtime: python37
```

The status bar at the bottom indicates the file is in 'master' branch, has 0 changes, and is using VML.

Step 9: Go to terminal and type the following command

python main.py

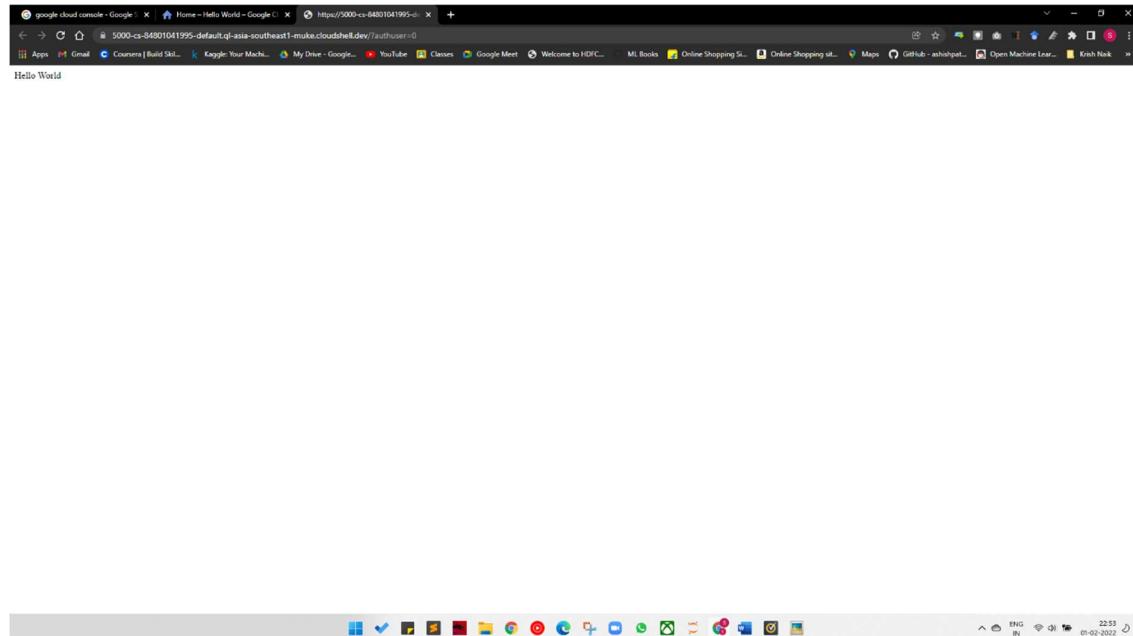


```
Welcome to Cloud Shell! Type 'help' to get started.
Your Cloud Platform project in this session is set to hello-world-340017.
Use "gcloud config set project [PROJECT_ID]" to change to a different project.
shashankadim12000@cloudshell:~$ gcloud config set project hello-world-340017& python main.py
shashankadim12000@cloudshell:~$ python main.py
property-finder python-docs-samples README-cloudshell.txt
shashankadim12000@cloudshell:~$ python main.py
-----
Python 2 is deprecated. Upgrade to Python 3 as soon as possible.
See https://cloud.google.com/python/docs/python3-upgrade
To suppress this warning, create an empty ~/.cloudshell/no-python-warning file.
The command will automatically process 1 seconds or on any key.
-----
* Service: flask app "main" (lazy loading)
Environment: production
This is a production WSGI server. Do not use it in a production deployment.
* Detected site: /var/www
* Running on http://127.0.0.1:15000/ (Press CTRL+C to quit)
* Documentation at: http://127.0.0.1:15000/docs/
* Debugger is active.
* Debugger PIN: B1B-016-713
127.0.0.1 - - [01-Feb-2022 17:22:36] "GET /authuser=Q HTTP/1.1" 200 -
127.0.0.1 - - [01-Feb-2022 17:22:37] "GET /favicon.ico HTTP/1.1" 404 -
```

Note: If Flask is not installed, please use the following command

pip install flask

Step 10: Open the https link



AMAZON WEB SERVICES

AMAZON WEB SERVICES

Amazon Web Services, Inc. (AWS) is a subsidiary of Amazon providing on-demand cloud computing platforms and APIs to individuals, companies, and governments, on a metered pay-as-you-go basis. These cloud computing web services provide a variety of basic abstract technical infrastructure and distributed computing building blocks and tools. One of these services is Amazon Elastic Compute Cloud (EC2), which allows users to have at their disposal a virtual cluster of computers, available all the time, through the Internet. AWS's virtual computers emulate most of the attributes of a real computer, including hardware central processing units (CPUs) and graphics processing units (GPUs) for processing; local/RAM memory; hard-disk/SSD storage; a choice of operating systems; networking; and pre-loaded application software such as web servers, databases, and customer relationship management (CRM).

The AWS technology is implemented at server farms throughout the world, and maintained by the Amazon subsidiary. Fees are based on a combination of usage (known as a "Pay-as-you-go" model), hardware, operating system, software, or networking features chosen by the subscriber required availability, redundancy, security, and service options. Subscribers can pay for a single virtual AWS computer, a dedicated physical computer, or clusters of either. As part of the subscription agreement, Amazon provides security for subscribers' systems. AWS operates from many global geographical regions including 6 in North America.

Amazon markets AWS to subscribers as a way of obtaining large scale computing capacity more quickly and cheaply than building an actual physical server farm. All services are billed based on usage, but each service measures usage in varying ways. As of 2017, AWS owns 33% of all cloud (IaaS, PaaS) while the next two competitors Microsoft Azure and Google Cloud have 18%, and 9% respectively, according to Synergy Group.

As of 2021, AWS comprises over 200 products and services including computing, storage, networking, database, analytics, application services, deployment, management, machine learning, mobile, developer tools, RobOps and tools for the Internet of Things. The most popular include Amazon Elastic Compute Cloud (EC2), Amazon Simple Storage Service (Amazon S3), Amazon Connect, and AWS Lambda (a serverless function enabling serverless ETL e.g., between instances of EC2 & S3).

Most services are not exposed directly to end users, but instead offer functionality through APIs for developers to use in their applications. Amazon Web Services' offerings are accessed over HTTP, using the REST architectural style and SOAP protocol for older APIs and exclusively JSON for newer ones.



Applications of AWS services

Amazon Web services are widely used for various computing purposes like:

- Web site hosting
- Application hosting/SaaS hosting
- Media Sharing (Image/ Video)
- Mobile and Social Applications
- Content delivery and Media Distribution
- Storage, backup, and disaster recovery
- Development and test environments
- Academic Computing
- Search Engines
- Social Networking

RELATIONAL DATABASES

Amazon Relational Database Service (Amazon RDS) makes it easy to set up, operate, and scale a relational database in the cloud. It provides cost-efficient and resizable capacity while automating time-consuming administration tasks, such as hardware provisioning, database setup, patching, and backups. It frees you to focus on your applications so you can give them the fast performance, high availability, security, and compatibility they need.

Amazon RDS is available on several database instance types - optimized for memory, performance, or I/O - and provides you with six familiar database engines to choose from, including Amazon Aurora, PostgreSQL, MySQL, MariaDB, Oracle Database, and SQL Server. You can use the AWS Database Migration Service to easily migrate or replicate your existing databases to Amazon RDS.



Benefits of Amazon RDS:

- Easy to Administer

Amazon RDS makes it easy to go from project conception to deployment. Use the Amazon RDS Management Console, the AWS RDS Command-Line Interface, or simple API calls to access the capabilities of a production-ready relational database in minutes. No need for infrastructure provisioning, and no need for installing and maintaining database software.

- Highly Scalable

You can scale your database's compute and storage resources with only a few mouse clicks or an API call, often with no downtime. Many Amazon RDS engine types allow you to launch one or more Read Replicas to offload read traffic from your primary database instance.

- Available and Durable

Amazon RDS runs on the same highly reliable infrastructure used by other Amazon Web Services. When you provision a Multi-AZ DB Instance, Amazon RDS synchronously replicates the data to a standby instance in a different Availability Zone (AZ). Amazon RDS has many other features that enhance reliability for critical production databases, including automated backups, database snapshots, and automatic host replacement.

- Fast

Amazon RDS supports the most demanding database applications. You can choose between two SSD-backed storage options: one optimized for high-performance OLTP applications, and the other for cost-effective general-purpose use. In addition, Amazon Aurora provides performance on par with commercial databases at 1/10th the cost.

- Secure

Amazon RDS makes it easy to control network access to your database. Amazon RDS also lets you run your database instances in Amazon Virtual Private Cloud (Amazon VPC), which enables you to isolate your database instances and to connect to your existing IT infrastructure through an industry-standard encrypted IPsec VPN. Many Amazon RDS engine types offer encryption at rest and encryption in transit.

- Inexpensive

You pay very low rates and only for the resources you actually consume. In addition, you benefit from the option of On-Demand pricing with no up-front or long-term commitments or even lower hourly rates via our Reserved Instance pricing.

VPC

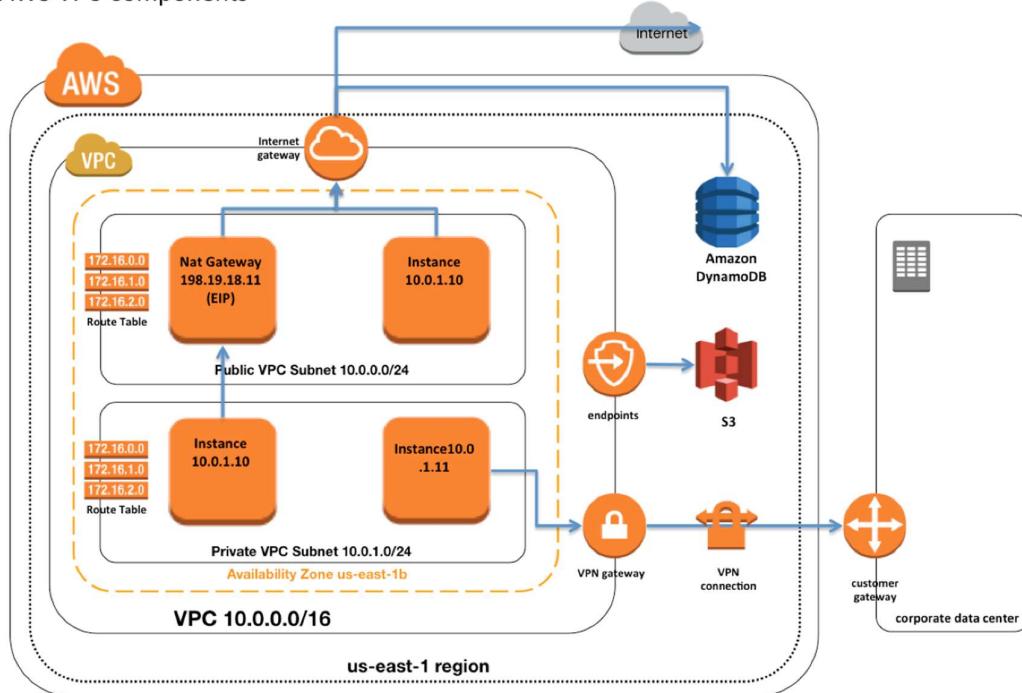
Amazon Virtual Private Cloud (Amazon VPC) enables you to launch AWS resources into a virtual network that you've defined. This virtual network closely resembles a traditional network that you'd operate in your own data centre, with the benefits of using the scalable infrastructure of AWS. Amazon VPC is the networking layer for Amazon EC2.

The following are the key concepts for VPCs:

- Virtual private cloud (VPC) — A virtual network dedicated to your AWS account.
- Subnet — A range of IP addresses in your VPC.
- CIDR block —Classless Inter-Domain Routing. An internet protocol address allocation and route aggregation methodology. For more information, see Classless Inter-Domain Routing in Wikipedia.

- Route table — A set of rules, called routes, that are used to determine where network traffic is directed.
- DHCP options sets: Configuration information (such as domain name and domain name server) passed to EC2 instances when they are launched into VPC subnets.
- Internet gateway — A gateway that you attach to your VPC to enable communication between resources in your VPC and the internet.
- Egress-only internet gateways: A type of internet gateway that allows an EC2 instance in a subnet to access the internet but prevents resources on the internet from initiating communication with the instance.
- VPC endpoint — Enables you to privately connect your VPC to supported AWS services and VPC endpoint services powered by PrivateLink without requiring an internet gateway, NAT device, VPN connection, or AWS Direct Connect connection. Instances in your VPC do not require public IP addresses to communicate with resources in the service.
- NAT gateway: A managed AWS service that allows EC2 instances in private subnets to connect to the internet, other VPCs, or on-premises networks.
- NAT instance: An EC2 instance in a public subnet that allows instances in private subnets to connect to the internet, other VPCs, or on-premises networks.
- Carrier gateways: For subnets in Wavelength Zones, this type of gateway allows inbound traffic from a telecommunication carrier network in a specific location and outbound traffic to a telecommunication carrier network and the internet.
- Prefix lists: A collection of CIDR blocks that can be used to configure VPC security groups, VPC route tables, and AWS Transit Gateway route tables and can be shared with other AWS accounts using Resource Access Manager (RAM).
- Security groups: Acts as a virtual firewall to control inbound and outbound traffic for an AWS resource, such as an EC2 instance. Each VPC comes with a default security group, and you can create additional security groups. A security group can be used only in the VPC for which it's created.

AWS VPC Components

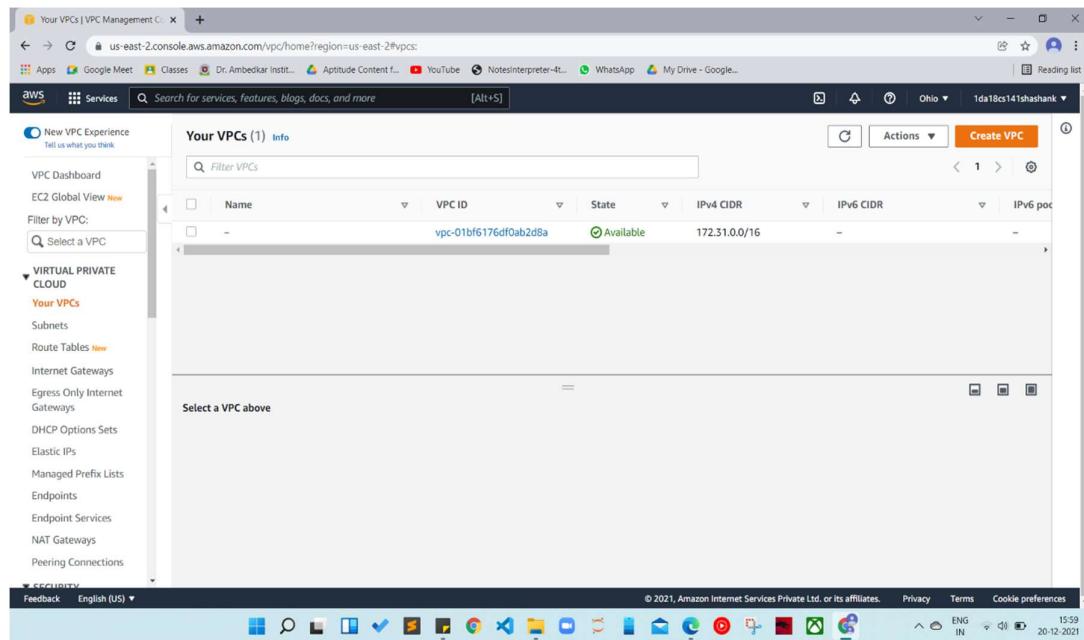


1.Create a RDS and launch in your custom VPC network.

Step 1: Open “AWS Management Console”. Click on “VPC” service.

Step 2: Click “Subnets” on the left panel.

Step 3: Now you can see there is one subnet group (Public Subnet) created in your VPC (Your VPC id/11.0.0.0/16). Now Click on “Create Subnet” button



Step 4: Give a Name to the subnet and select you own VPC from the “VPC drop down”. Select an “availability zone” (Which is not used by “Public Subnet” of your VPC). Give CIDR block range. Click on “Yes, Create” button.

Step 5: Now you can see the subnet is created in your VPC.

Step 6: Click on “Services”. Click on “RDS” service.

Step 7: Click on “Subnet Groups” on the left panel. (Note: Before creating the subnet groups you need to note down your VPC ID and subnets for that VPC).

Step 8: Click on “Create DB Subnet Group” button.

Step 9: Give a name to the “Subnet”. Select your own “VPC ID”. Select the “Availability Zone” and “subnet”. Click on “Add” button.

Step 10: Now select another “Availability Zone” and “Subnet”. Click on “Yes, Create” button. (Note: Before clicking on “Yes, create” check that two subnets are added or not)

Step 11: Now you can see that “DB subnet group” is created.

Step 12: Click on “RDS Dashboard” on the left panel. Click on “Get Started” button

Step 13: Select your desired “Database Engine” from the list and click on “Select” button.

Step 14: If you want multi-AZ Deployment, select the “first radio button” otherwise select the “second radio button”. Click on “Next” button.

Subnet ID	Subnet ARN	State	IPv4 CIDR
subnet-0b1bb051855021d47	arn:aws:ec2:us-east-2:881380715570:subnet/subnet-07c4baa3fbf019c49	Available	172.31.16.0/20
subnet-05421340c7af62103	arn:aws:ec2:us-east-2:881380715570:subnet/subnet-05421340c7af62103	Available	172.31.32.0/20
my-subnet-01	arn:aws:ec2:us-east-2:881380715570:subnet/subnet-07c4baa3fbf019c49	Available	172.31.64.0/24

Step 15: Select your desired DB instance from the “DB Instance Class” drop down. Select Multi-AZ option from the drop down. Select the storage type from the drop down and give 95 your desired storage space range. Enter the Details (DB instance Identifier, Master user name, password and confirm password). Click on “Next” button.

Subnet group details

- VPC ID: vpc-01bf6176df0ab2d8a
- ARN: arn:aws:rds:us-east-2:881380715570:subgrp:mydbsubnet
- Description: db-subnet

Subnets (4)

Availability zone	Subnet ID	CIDR block
us-east-2a	subnet-0819dfe5826c9b219	172.31.0.0/20
us-east-2b	subnet-0b1bb051855021d47	172.31.16.0/20
us-east-2c	subnet-05421340c7af62103	172.31.32.0/20
us-east-2a	subnet-07c4baaa3fb019c49	172.31.64.0/24

Step 16: Select your own VPC ID from the “VPC Dropdown”. Select the “public accessibility” from the dropdown. Select your desired Availability Zone from the dropdown. Give a name to your Database and check the database port is and mention the “Backup retention period” as per your needs and if you want a Time frame for your backup, configure the time frame from “Backup Window” Dropdown. Click on “Launch DB Instance” button.

Successfully created Mydbsubnet. View subnet group

Subnet groups (1)

Name	Description	Status	VPC
mydbsubnet	db-subnet	Complete	vpc-01bf6176df0ab2d8a

Screenshots illustrating the creation of a MySQL database on the AWS RDS Management Console.

The first screenshot shows the "Create database" wizard. Under "Choose a database creation method", "Standard create" is selected. In the "Engine options" section, "MySQL" is chosen. A sidebar on the right provides information about MySQL, stating it is the most popular open-source database and listing its features:

- Supports database size up to 64 TiB.
- Supports General Purpose, Memory Optimized, and Burstable Performance instance classes.
- Supports automated backup and point-in-time recovery.
- Supports up to 5 Read Replicas per instance, within a single Region or cross-region.

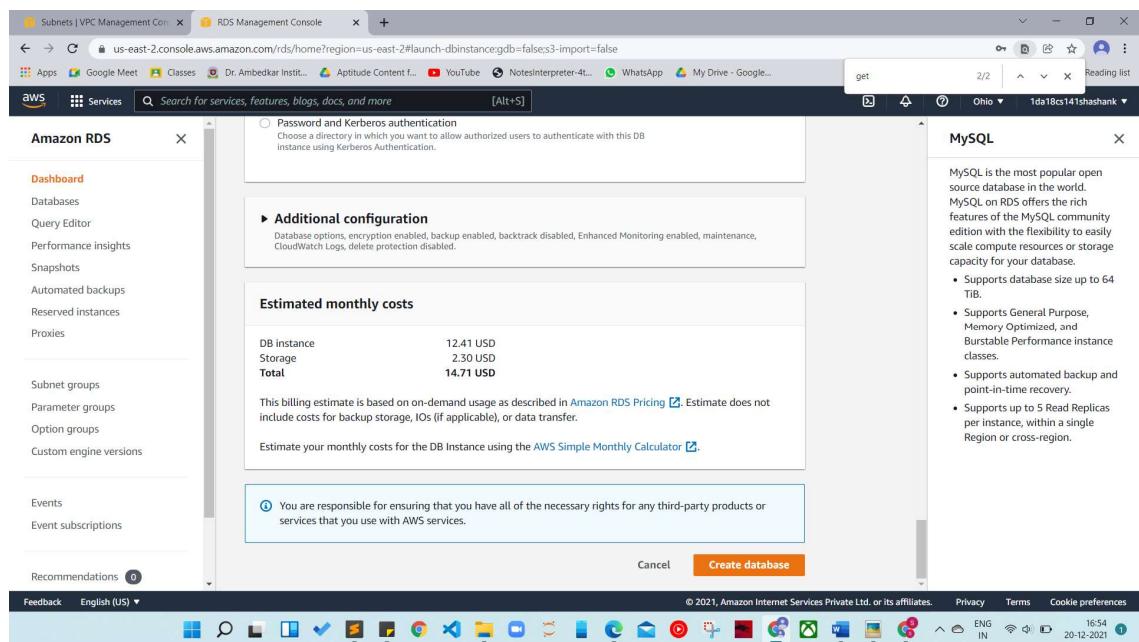
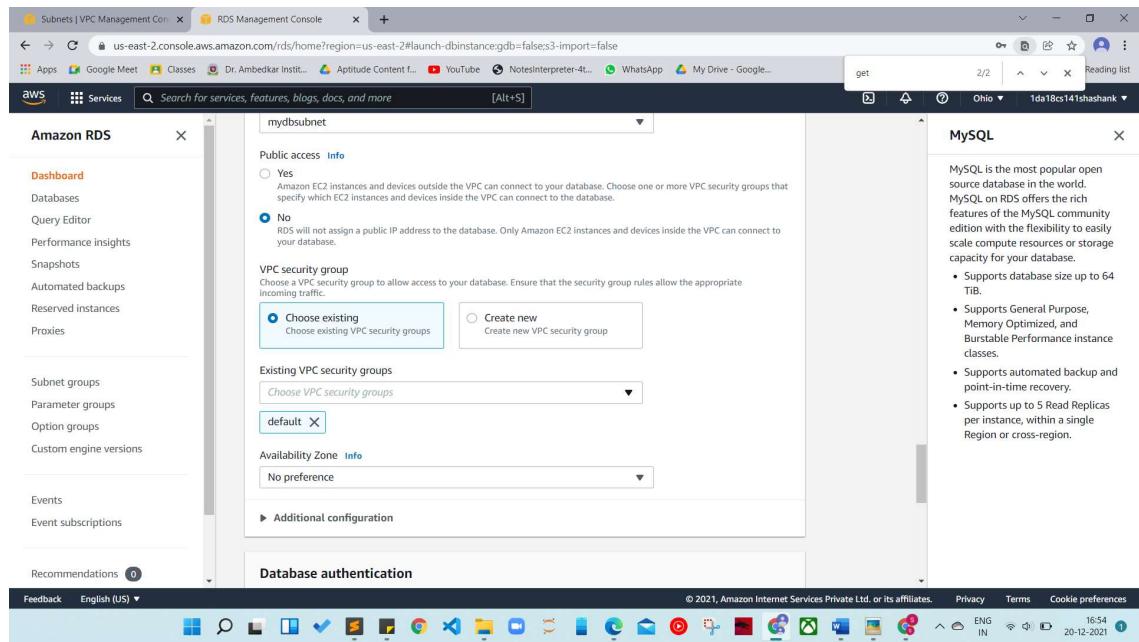
The second screenshot shows the continuation of the "Create database" wizard. The "Edition" dropdown is set to "MySQL Community". A "Known issues/limitations" box contains a note: "Review the Known issues/limitations [link] to learn about potential compatibility issues with specific database versions." The "Version" dropdown is set to "MySQL 8.0.23".

The screenshot shows the AWS RDS Management Console. On the left, a sidebar menu for 'Amazon RDS' includes options like Dashboard, Databases, Query Editor, Performance insights, Snapshots, Automated backups, Reserved instances, Proxies, Subnet groups, Parameter groups, Option groups, Custom engine versions, Events, Event subscriptions, and Recommendations. The main content area is titled 'Settings' under 'MySQL'. It contains fields for 'DB instance identifier' (Mydb), 'Master username' (Shashank), 'Master password' (*****), and 'Confirm password' (*****). A note states: 'The DB instance identifier is case-insensitive, but is stored as all lowercase (as in "mydbinstance"). Constraints: 1 to 60 alphanumeric characters or hyphens. First character must be a letter. Can't contain two consecutive hyphens. Can't end with a hyphen.' To the right, a panel titled 'MySQL' provides information about the service, including its popularity and features such as support for up to 64 TiB, General Purpose, Memory Optimized, and Burstable Performance instance classes, automated backup, point-in-time recovery, and up to 5 Read Replicas per instance.

This screenshot shows the 'Create New DB Instance' configuration page. The left sidebar is identical to the previous screenshot. The main area has a 'Confirm password' field (*****). Below it is the 'DB instance class' section, which includes a dropdown for 'DB instance class info' (selected: db.t3.micro) and an option to 'Include previous generation classes'. The 'Storage' section includes a dropdown for 'Storage type info' (General Purpose SSD (gp2)) and a 'Allocated storage' input field set to 20 GiB. The right panel, titled 'MySQL', reiterates the service's popularity and its features.

The screenshot shows the AWS RDS Management Console for MySQL. On the left, the sidebar includes options like Dashboard, Databases, Query Editor, Performance insights, Snapshots, Automated backups, Reserved instances, Proxies, Subnet groups, Parameter groups, Option groups, Custom engine versions, Events, Event subscriptions, and Recommendations. The main panel is titled 'Storage' and contains settings for 'General Purpose SSD (gp2)'. It shows an 'Allocated storage' of 20 GiB (minimum 20 GiB, maximum 16,384 GiB). A note states that provisioning less than 100 GiB could result in higher latencies. Under 'Storage autoscaling', the 'Enable storage autoscaling' checkbox is checked. A 'Maximum storage threshold' of 1000 GiB is set (minimum 21 GiB, maximum 16,384 GiB). To the right, a sidebar provides information about MySQL, listing its features such as support for up to 64 TiB and up to 5 Read Replicas per instance.

This screenshot shows the 'Availability & durability' section of the AWS RDS Management Console for MySQL. It includes options for 'Multi-AZ deployment' where the 'Do not create a standby instance' radio button is selected. In the 'Connectivity' section, the 'Virtual private cloud (VPC)' dropdown is set to 'Default VPC (vpc-01bf6176df0ab2d8a)', and the 'Subnet group' dropdown is set to 'mydbsubnet'. A note indicates that VPCs must have a corresponding DB subnet group. The right sidebar continues to provide MySQL details, including its popularity and various instance classes.



Step 17: Click on “Close” button.

Step 18: Now you can see one DB instance is created. (It will take 5-10min to create the database instance). You will have the DB name, VPC ID, End Point (which is used to connect to the DB Instance from your EC2 instance) and etc...

Creating database mydb
Your database might take a few minutes to launch.

RDS > Databases

DB identifier	Role	Engine	Region & AZ	Size	Status	CPU
mydb	Instance	MySQL Community	us-east-2c	db.t3.micro	Backing-up	-

Successfully created database mydb

RDS > Databases

DB identifier	Role	Engine	Region & AZ	Size	Status	CPU
mydb	Instance	MySQL Community	us-east-2a	db.t2.micro	Available	3.8%

The screenshot shows the AWS RDS Management Console interface. A success message at the top states "Successfully created database mydb". The main area displays the "mydb" database details under the "Databases" section. The "Summary" tab is selected, showing the following information:

DB identifier	CPU	Status	Class
mydb	3.87%	Available	db.t2.micro
Role	Current activity	Engine	Region & AZ
Instance	0 Connections	MySQL Community	us-east-2a

Below the summary, there are tabs for "Connectivity & security", "Monitoring", "Logs & events", "Configuration", "Maintenance & backups", and "Tags". The "Connectivity & security" tab is active, showing the endpoint information:

Endpoint & port	Networking	Security
Endpoint mydb.cimlmctuy6d.us-east-2.rds.amazonaws.com	Availability Zone us-east-2a	VPC security groups default (sg-0898a58672ebef9f4) Active

Note: To connect to the Database from your Ec2 Instance, you need the following.

- a). RDS end point.
- b) Database Name.
- c) Master username.
- d) Master Password.
- e) Port Number.

VM WARE, VIRTUAL BOX AND GUEST OS

Virtual Box

VirtualBox is a powerful x86 and AMD64/Intel64 virtualization product for enterprise as well as home use. Not only is VirtualBox an extremely feature rich, high-performance product for enterprise customers, it is also the only professional solution that is freely available as Open-Source Software under the terms of the GNU General Public License (GPL) version 2. See "About VirtualBox" for an introduction.

Presently, VirtualBox runs on Windows, Linux, Macintosh, and Solaris hosts and supports a large number of guest operating systems including but not limited to Windows (NT 4.0, 2000, XP, Server 2003, Vista, Windows 7, Windows 8, Windows 10), DOS/Windows 3.x, Linux (2.4, 2.6, 3.x and 4.x), Solaris and Open Solaris, OS/2, and OpenBSD.

VirtualBox is being actively developed with frequent releases and has an ever-growing list of features, supported guest operating systems and platforms it runs on. VirtualBox is a community effort backed by a dedicated company: everyone is encouraged to contribute while Oracle ensures the product always meets professional quality criteria.



History:

VirtualBox was first offered by Innotek GmbH from Weinstadt, Germany, under a proprietary software license, making one version of the product available at no cost for personal or evaluation use, subject to the VirtualBox Personal Use and Evaluation License (PUEL). In January 2007, based on counsel by LiSoG, Innotek GmbH released VirtualBox Open-Source Edition (OSE) as free and open-source software, subject to the requirements of the GNU General Public License (GPL), version 2. Innotek GmbH also contributed to the development of OS/2 and Linux support in virtualization and OS/2 ports of products from Connectix which were later acquired by Microsoft. Specifically, Innotek developed the "additions" code in both Windows Virtual PC and Microsoft Virtual Server, which enables various host–guest OS interactions like shared clipboards or dynamic viewport resizing.

Sun Microsystems acquired Innotek in February 2008. Following the acquisition of Sun Microsystems by Oracle Corporation in January 2010, the product was re-branded as "Oracle VM VirtualBox". In December 2019, VirtualBox started supporting only hardware-assisted virtualization, dropping support for Software-based one.

Features:

- Snapshots of the RAM and storage that allow reverting to a prior state.
- Screenshots and screen video capture
- "Host key" for releasing the keyboard and mouse cursor to the host system if captured (coupled) to the guest system, and for keyboard shortcuts to features such as configuration, restarting, and screenshot. By default, it is the right-side CTRL key.
- Mouse pointer integration, meaning automatic coupling and uncoupling of mouse cursor when moved inside and outside the virtual screen, if supported by guest operating system.
- Seamless mode – the ability to run virtualized applications side by side with normal desktop applications
- Shared clipboard
- Shared folders through "guest additions" software
- Special drivers and utilities to facilitate switching between systems

- Ability to specify amount of shared RAM, video memory, and CPU execution cap
- Ability to emulate multiple screens
- Command line interaction (in addition to the GUI)
- Public API (Java, Python, SOAP, XPCOM) to control VM configuration and execution
- Nested paging for AMD-V and Intel VT (only for processors supporting SLAT and with SLAT enabled)
- Limited support for 3D graphics acceleration (including OpenGL up to (but not including) 3.0 and Direct3D 9.0c via Wine's Direct3D to OpenGL translation)
- SMP support (up to 32 virtual CPUs per virtual machine), since version 3.0
- Teleportation (aka Live Migration)
- 2D video output acceleration (not to be mistaken with video decoding acceleration), since version 3.1
- EFI has been supported since version 3.1 (Windows 7 guests are not supported).

VM Ware

VMware, Inc. is an American cloud computing and virtualization technology company with headquarters in California. VMware was the first commercially successful company to virtualize the x86 architecture. VMware's desktop software runs on Microsoft Windows, Linux, and macOS, while its enterprise software hypervisor for servers, VMware ESXi, is a bare-metal hypervisor that runs directly on server hardware without requiring an additional underlying operating system. VMware's most notable products are its hypervisors. VMware became well known for its first type 2 hypervisor known as GSX. This product has since evolved into two hypervisor product lines: VMware's type 1 hypervisors running directly on hardware and their hosted type 2 hypervisors. VMware software provides a completely virtualized set of hardware to the guest operating system. VMware software virtualizes the hardware for a video adapter, a network adapter, and hard disk adapters. The host provides pass-through drivers for guest USB, serial, and parallel devices. In this way, VMware virtual machines become highly portable between computers, because every host looks nearly identical to the guest. In practice, a system administrator can pause operations on a virtual machine guest,

move or copy that guest to another physical computer, and their resume execution exactly at the point of suspension. Alternatively, for enterprise servers, a feature called vMotion allows the migration of operational guest virtual machines between similar but separate hardware hosts sharing the same storage (or, with vMotion Storage, separate storage can be used, too). Each of these transitions is completely transparent to any users on the virtual machine at the time it is being migrated.

VMware's products predate the virtualization extensions to the x86 instruction set, and do not require virtualization-enabled processors. On newer processors, the hypervisor is now designed to take advantage of the extensions. However, unlike many other hypervisors, VMware still supports older processors. In such cases, it uses the CPU to run code directly whenever possible (as, for example, when running user-mode and virtual 8086 mode code on x86). When direct execution cannot operate, such as with kernel-level and real-mode code, VMware products use binary translation (BT) to re-write the code dynamically. The translated code gets stored in spare memory, typically at the end of the address space, which segmentation mechanisms can protect and make invisible. For these reasons, VMware operates dramatically faster than emulators, running at more than 80% of the speed that the virtual guest operating system would run directly on the same hardware. In one study VMware claims a slowdown over native ranging from 0–6 percent for the VMware ESX Server.

Products:

Desktop software

- VMware Workstation, introduced in 1999, was the first product launched by VMware. This software suite allows users to run multiple instances of x86 or x86-64 -compatible operating systems on a single physical personal computer. Workstation Pro version 15.5.1 was released in Nov 2019.
- VMware Fusion provides similar functionality for users of the Intel Mac platform, along with full compatibility with virtual machines created by other VMware products.
- VMware Workstation Player is freeware for non-commercial use, without requiring a license, and available for commercial use with permission. It is similar to VMware Workstation, with reduced functionality.

Server software

- VMware ESXi, an enterprise software product, can deliver greater performance than the freeware VMware Server, due to lower system computational overhead. VMware ESXi, as a "bare-metal" product, runs directly on the server hardware, allowing virtual servers to also use hardware more or less directly. In addition, VMware ESXi integrates into VMware vCenter, which offers extra services.

Cloud management software

- VMware vRealize Suite – a cloud management platform purpose-built for a hybrid cloud.
- VMware Go is a web-based service to guide users of any expertise level through the installation and configuration of VMware vSphere Hypervisor.
- VMware Cloud Foundation – Cloud Foundation provides an easy way to deploy and operate a private cloud on an integrated SDDC system.
- VMware Horizon View is a virtual desktop infrastructure (VDI) product.

Application management

- The VMware Workspace Portal was a self-service app store for workspace management.

Storage and availability

VMware's storage and availability products are composed of two primary offerings:

- VMware vSAN (previously called VMware Virtual SAN) is software-defined storage that is embedded in VMware's ESXi hypervisor. The vSphere and vSAN software run on industry-standard x86 servers to form a hyper-converged infrastructure (or HCI). However, network operators need to have servers from HCL (Hardware Compatibility List) to put one into production. The first release, version 5.5, was released in March 2014. The 6th generation, version 6.6, was released in April 2017. New features available in VMware vSAN 6.6 include native data at rest encryption, local protection for stretched clusters, analytics, and optimized solid-state drive performance. The VMWare 6.7 version was released in April 2018, Users now have improved monitoring tools and new workflows, it is closer to feature parity. The vCenter Server Appliance architecture is moving around to an easy deployment method.

- VMware Site Recovery Manager (SRM) automates the failover and failback of virtual machines to and from a secondary site using policy-based management.

Networking and security products

- VMware NSX is VMware's network virtualization product marketed using the term software-defined data centre (SDDC). The technology included some acquired from the 2012 purchase of Nicira. Software Defined Networking (SDN) allows the same policies that govern Identity and Access Management (IAM) to dictate levels of access to applications and data through a totally converged infrastructure not possible with legacy network and system access methods.

Other products

- Workspace ONE allows mobile users to access apps and data.
- The VIX (Virtual Infrastructure eXtension) API allows automated or scripted management of a computer virtualized using either VMware's vSphere, Workstation, Player, or Fusion products. VIX provides bindings for the programming languages C, Perl, Visual Basic, VBscript and C#.
- Herald is an communications protocol from VMware for more reliable Bluetooth communication and range finding across for mobile devices. Herald code is available under an Open-source license and was implemented in the Australian Government's COVID Safe app for contact tracing on 19 December 2020.

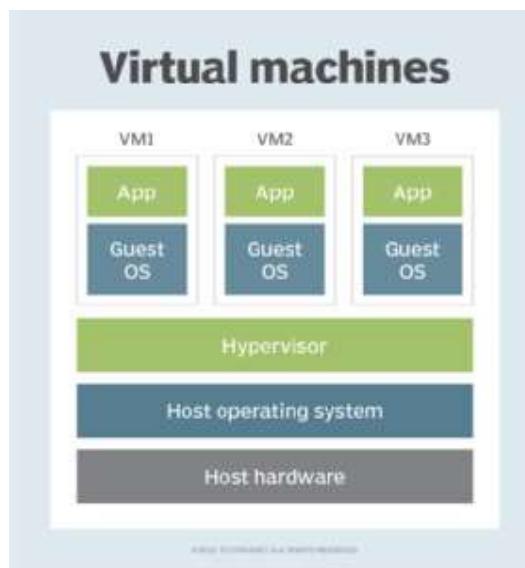


Guest OS

A guest operating system (guest OS) is an operating system (OS) that is secondary to the OS originally installed on a computer, which is known as the host operating system. The guest OS is either part of a partitioned system or part of a virtual machine (VM) setup. A guest OS provides an alternative OS for a device.

In disk partitioning, a guest OS is simply another instance of the same operating system that can boot up for controlling a certain partitioned memory set. A virtual machine (VM) process is much different, in that a guest OS can be a different OS alternative. In VM setups, a guest OS is delivered through a virtual machine environment through a tool called a hypervisor. Again, the machine will typically have a host OS, where the guest OS will operate "within" the host OS. This can lead to limitations on file saving and other operations within the guest OS, depending on whether the guest OS is said to be "persistent."

Part of the emergence of guest operating systems in VM systems has to do with the benefits provided by virtualization. These revolutions in computing coincide with the more general concept of cloud computing, where resources are delivered, rather than hosted, in physical local hardware setups. In addition, a guest OS often takes advantage of a lean OS build, where memory requirements are further alleviated. VM setups can help with licensing issues, system requirements and more, making these an attractive part of an outsourced computing service.



HOST OPERATING SYSTEM VERSUS GUEST OPERATING SYSTEM

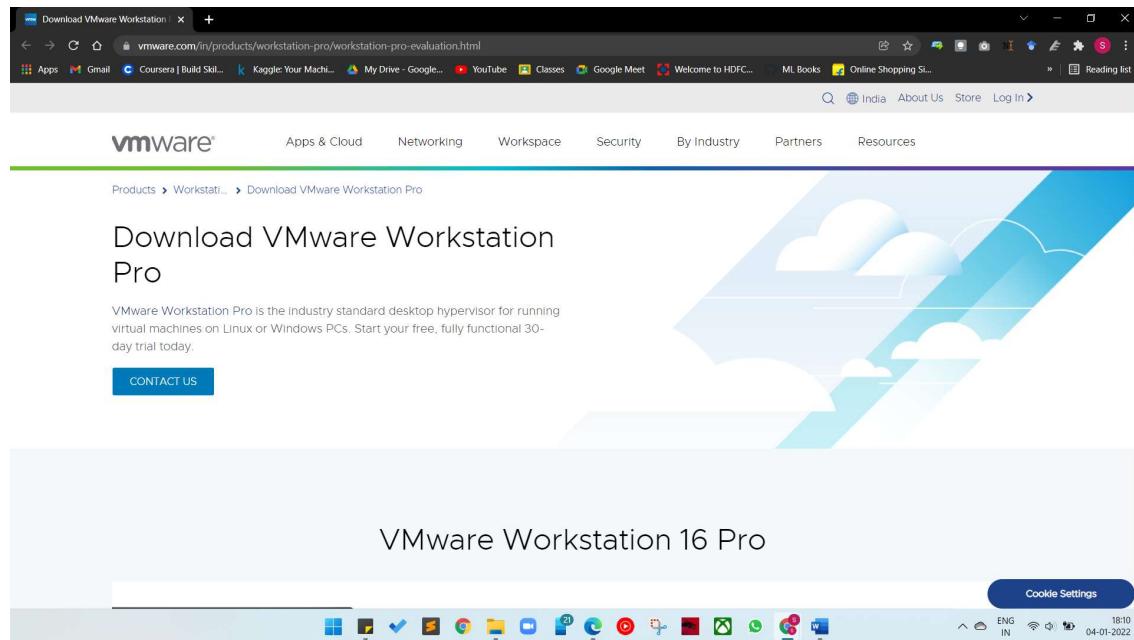
HOST OPERATING SYSTEM	GUEST OPERATING SYSTEM
A software installed on a computer that interacts with the underlying hardware	A software installed in a virtual machine
Host OS interacts on the hardware	Guest OS runs on a virtual machine
There is a single host operating system	There can be a single or multiple guest operating systems

Visit www.PEDIAA.com

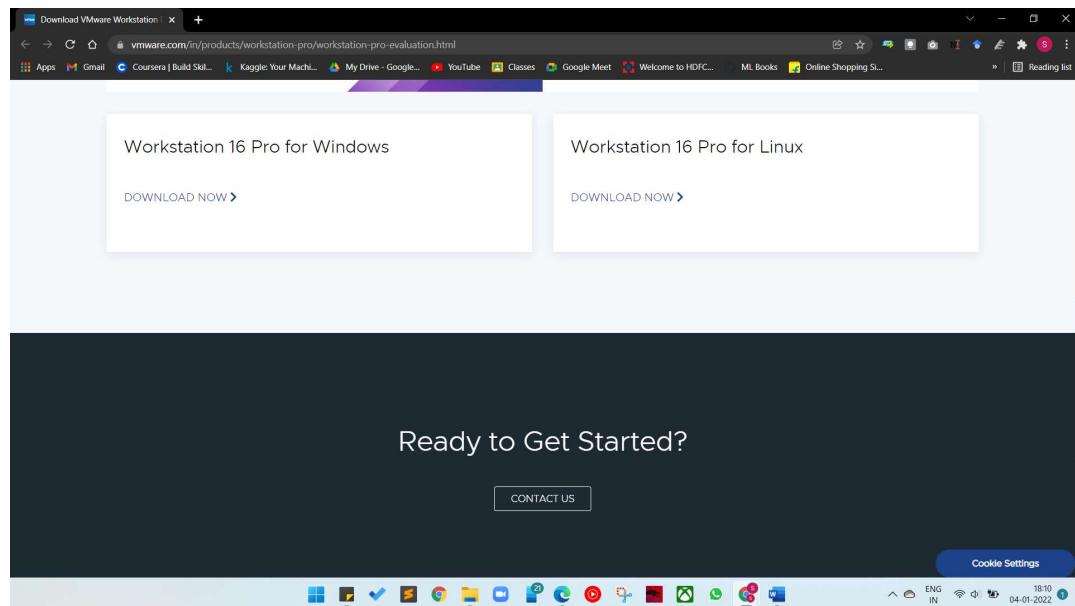
1. Install VirtualBox/VMware Workstation with different flavours of Linux and execute some C programs

Step 1: To download and install the VMware product visit the official website of VMware.

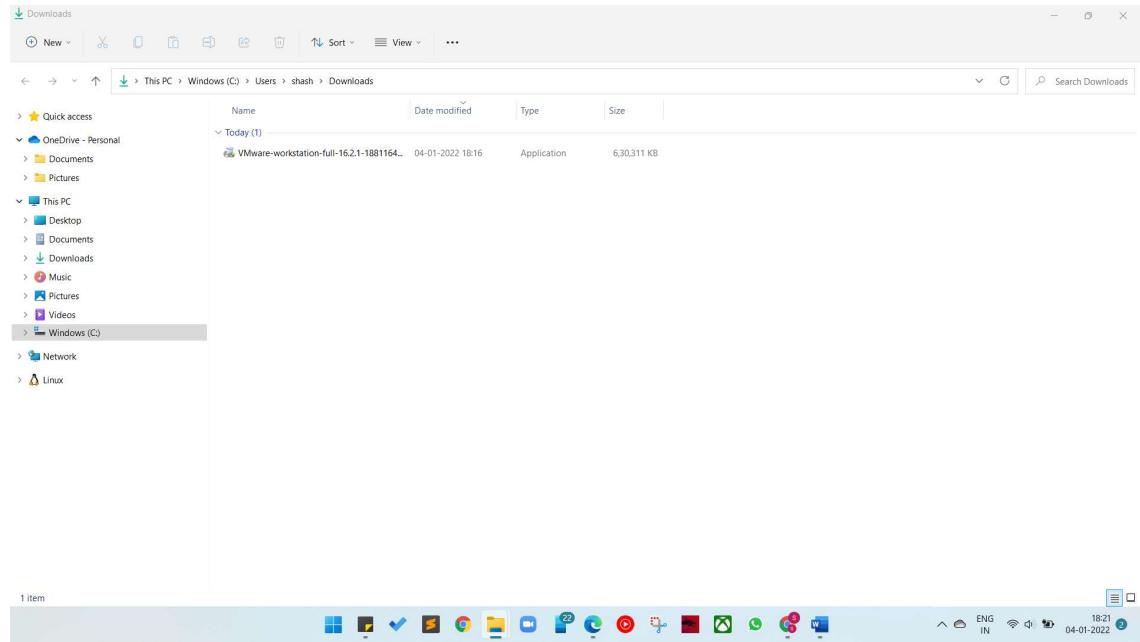
<https://www.vmware.com/in/products/workstation-pro/workstationproevaluation.html>



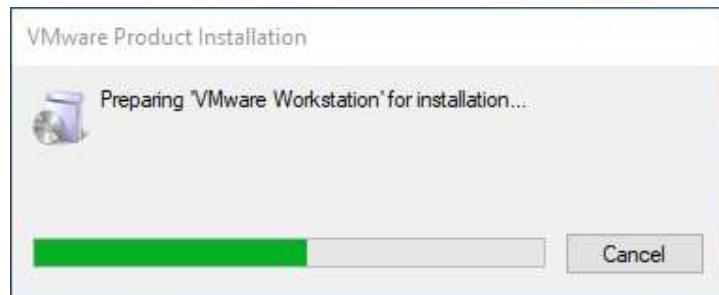
Step 2: Click on Download VM WorkStation for Windows.



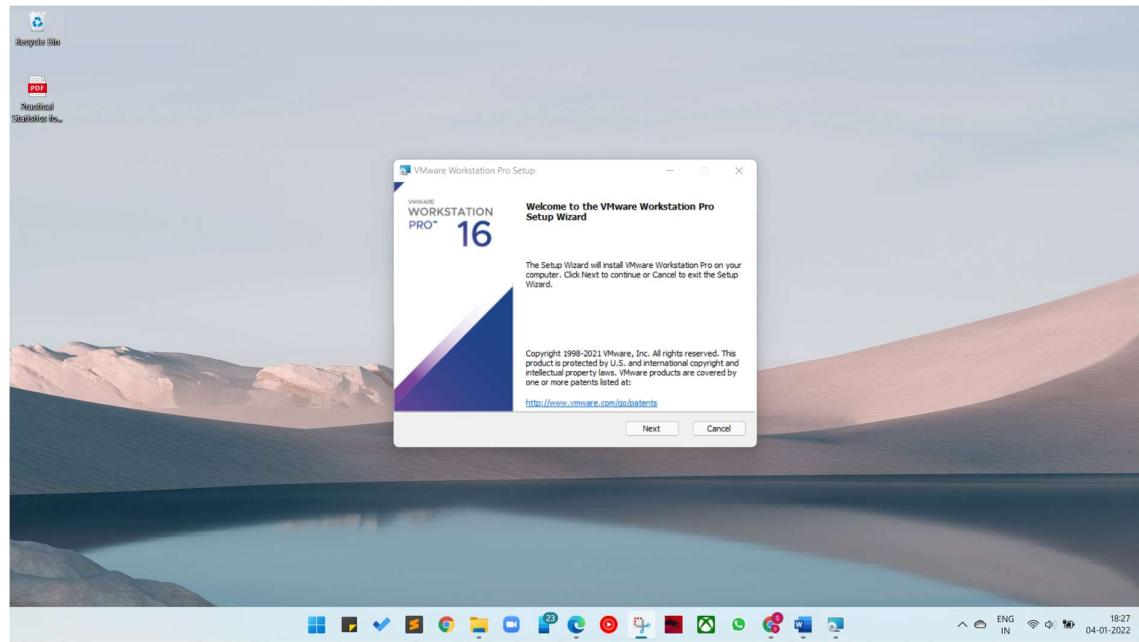
The installation file gets downloaded in the specified location and is now ready for installation.



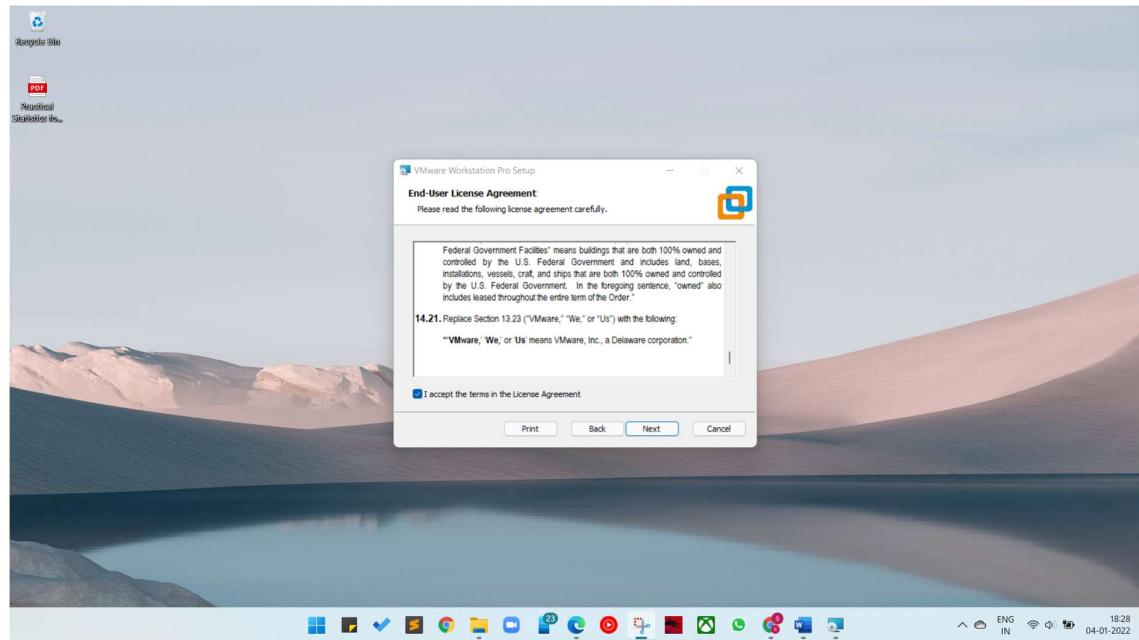
Step 3: Click on the download file to install the VMWare Workstation 16 Pro.
Popup will appear



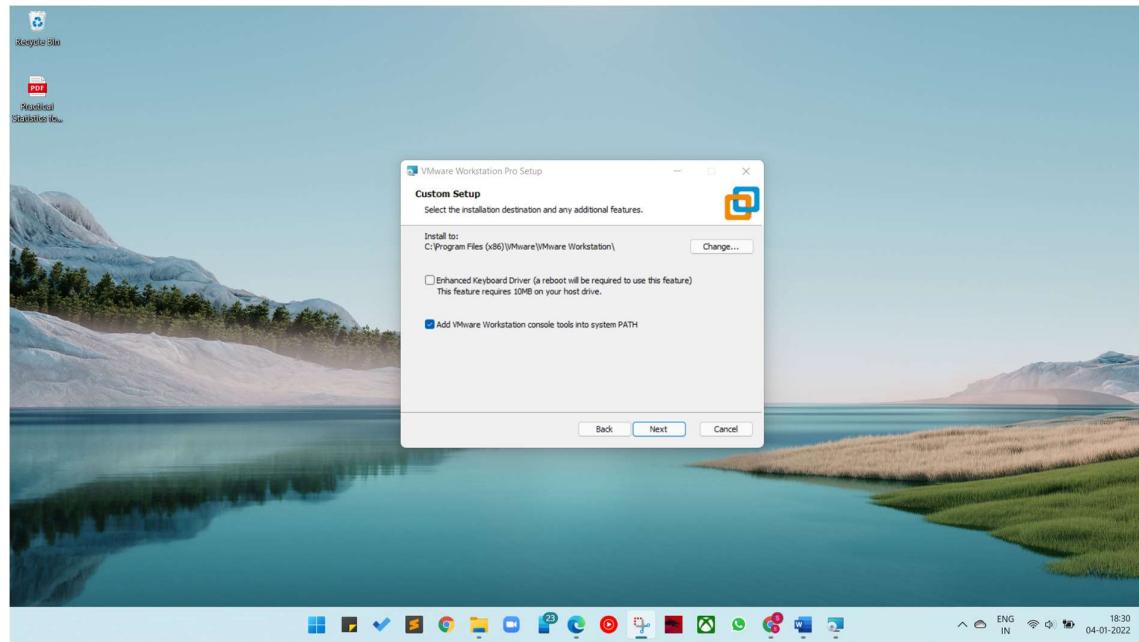
Step 4: Once the initialization gets completed, Click Next



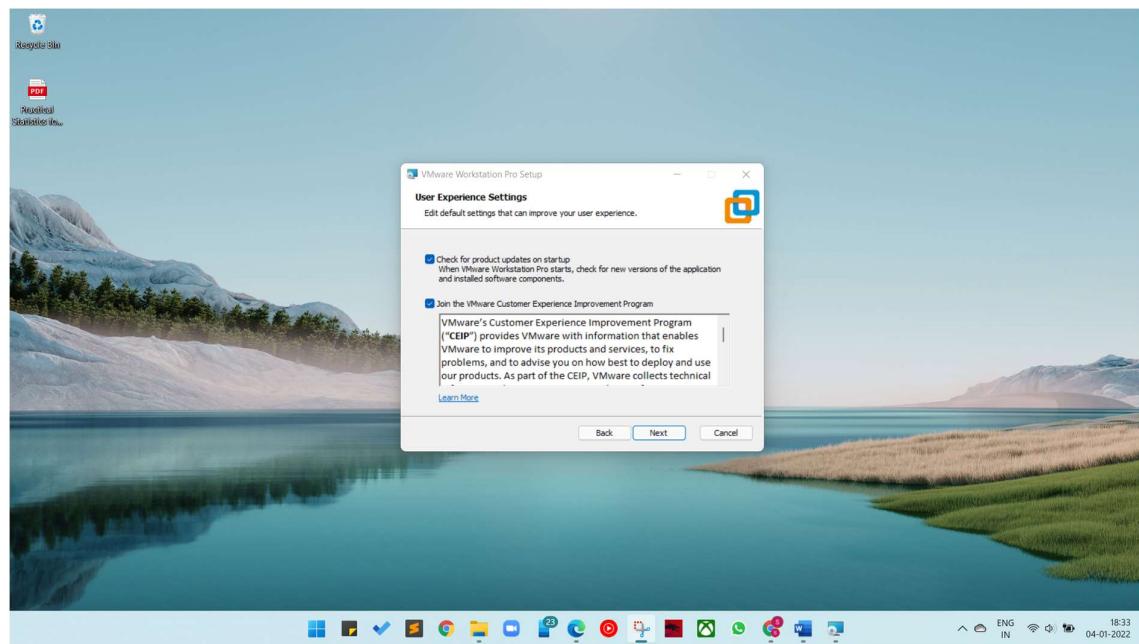
Step 5: Accept the terms and click on Next

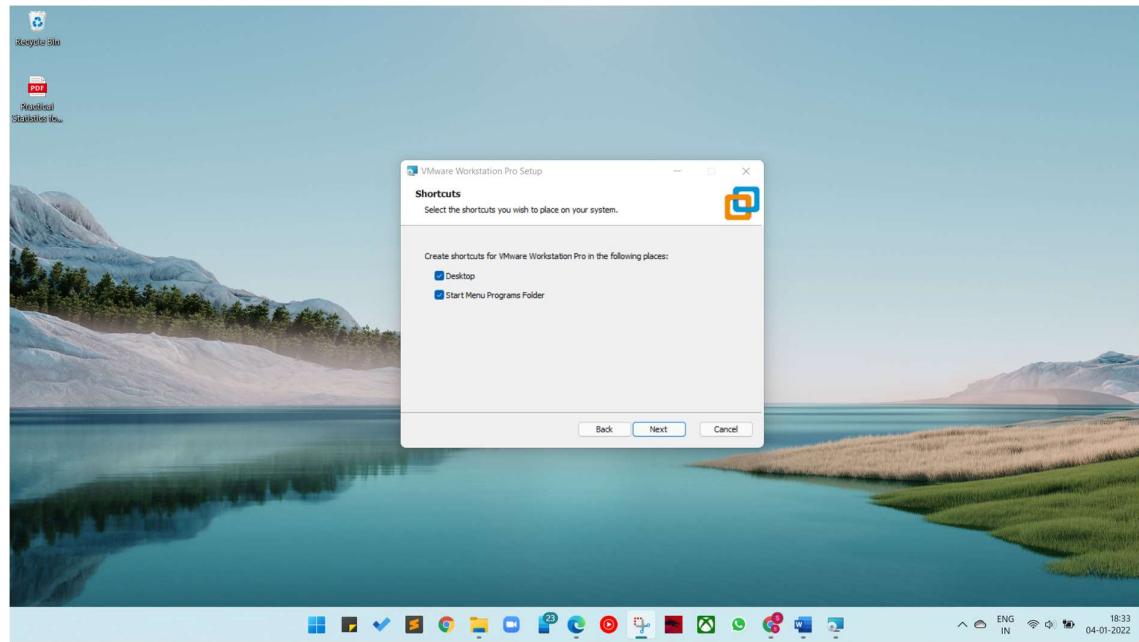


In the next screen, it will ask for some additional features, it is not mandatory to check this box. Click on Next.

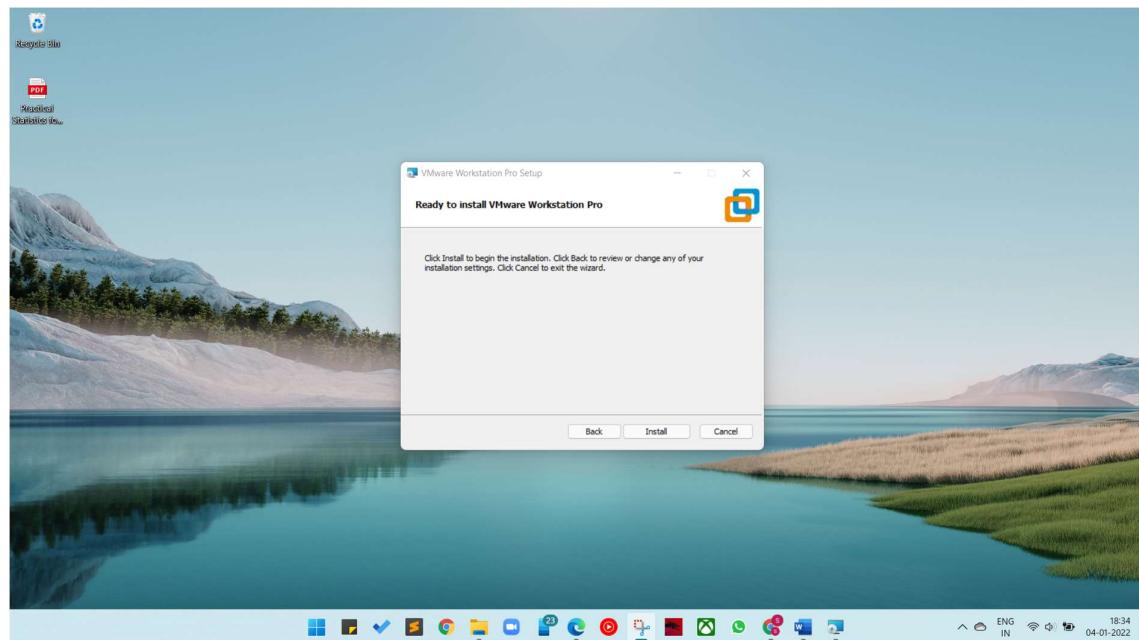


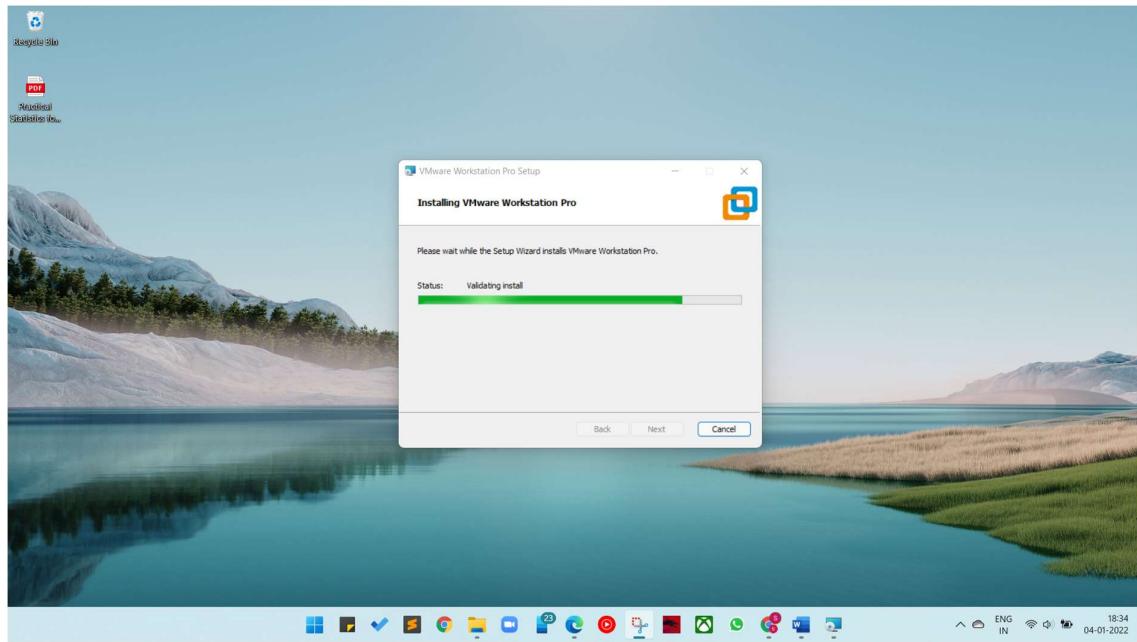
Step 6: On the next screen, some checkboxes are populated, Check them as per your requirement. Click on Next.



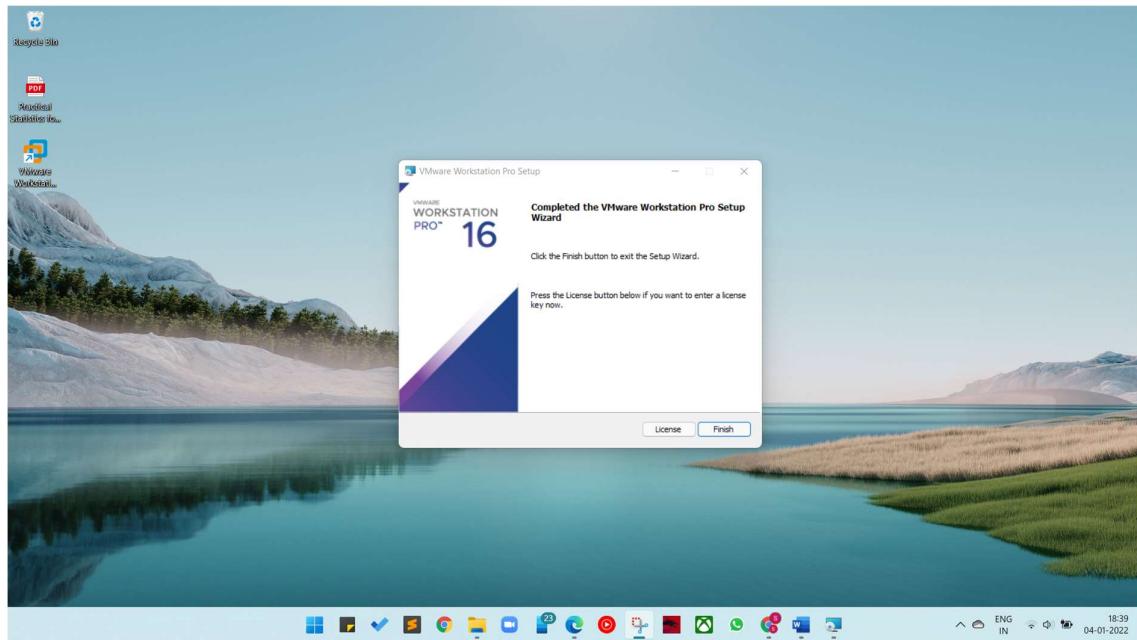


Step 7: At this step, VMware Workstation is ready to install. Click on Install.



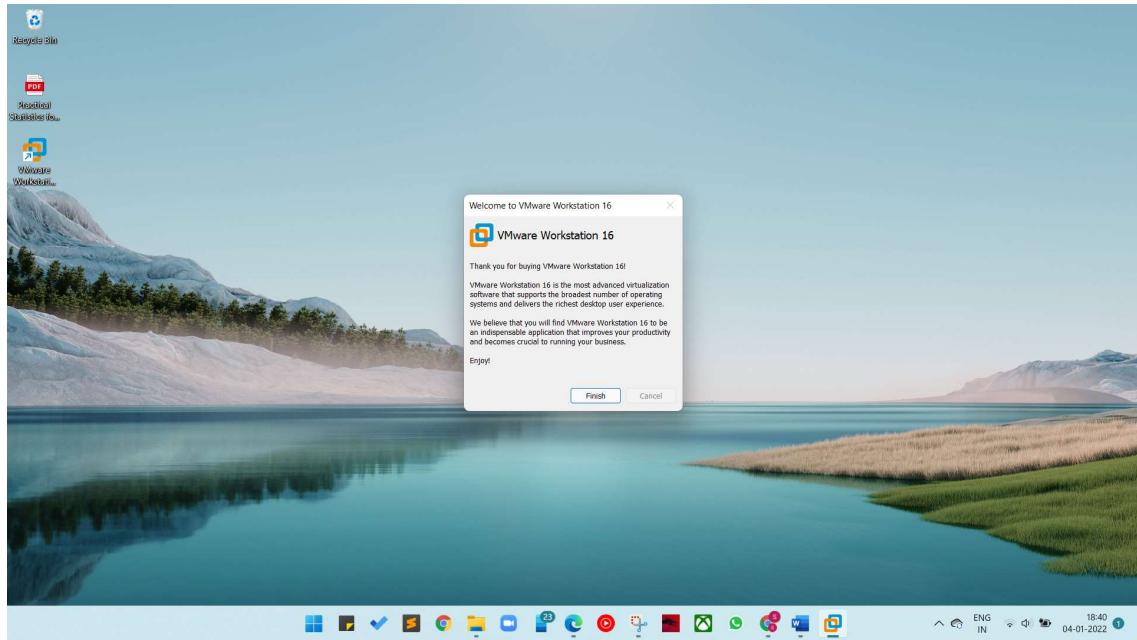
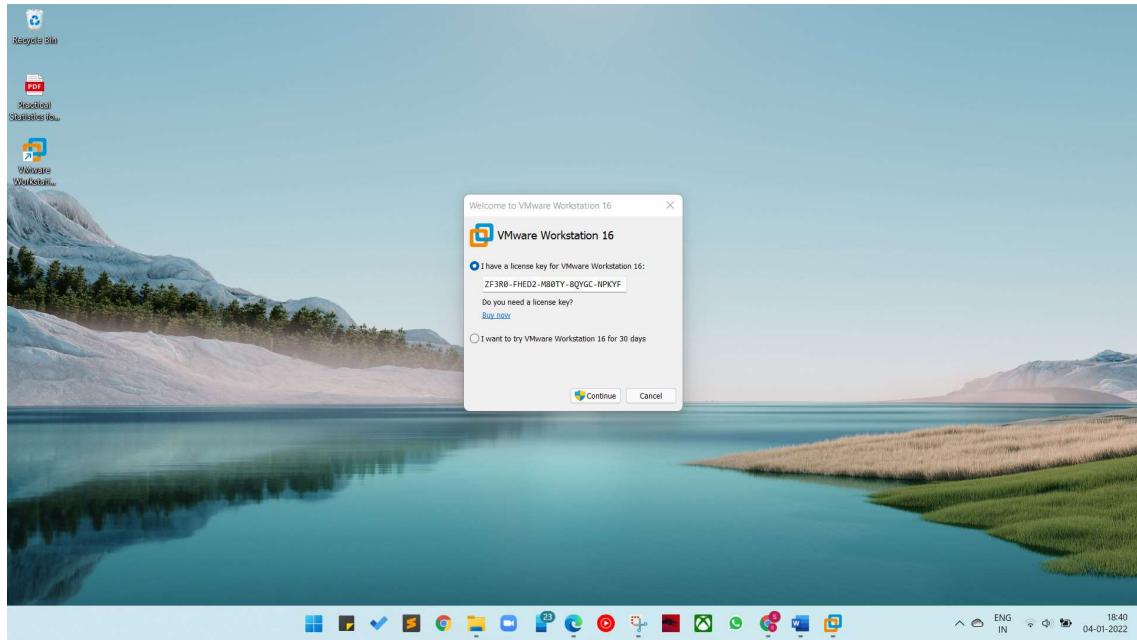


Step 8: Once the installation gets completed you will see the following dialogue box. Click on Finish. If you have purchased the product and have a license key, then you can click on License to enter the key.

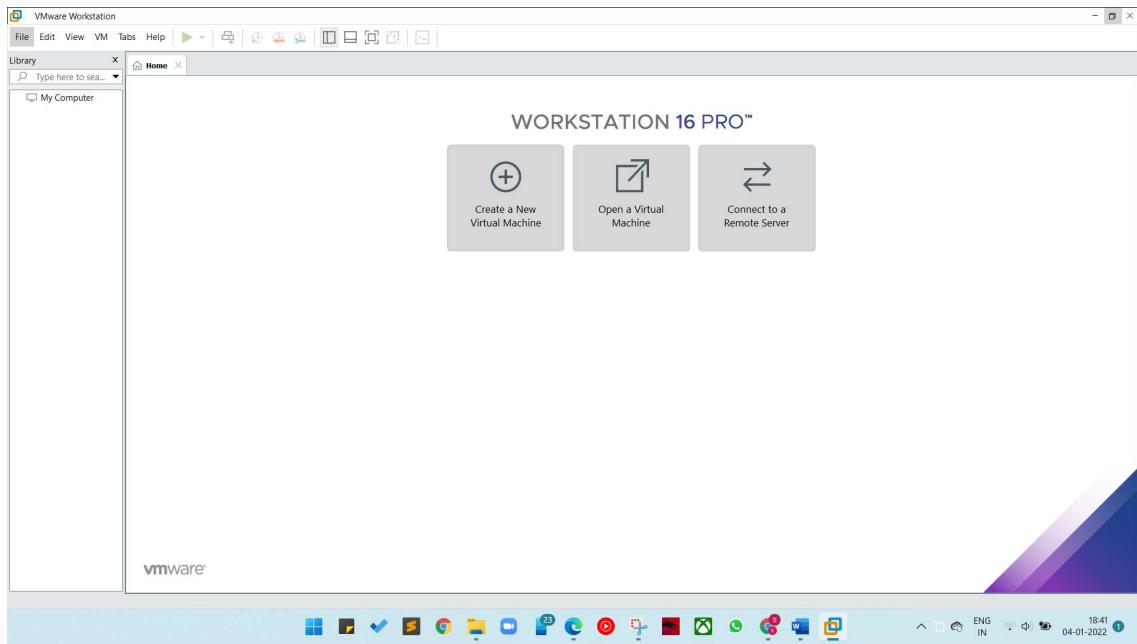


Step 9: Upon Finish, the window will close, and You can see VMware Workstation installed icon on your Desktop. Double Click on the Icon to open the application.

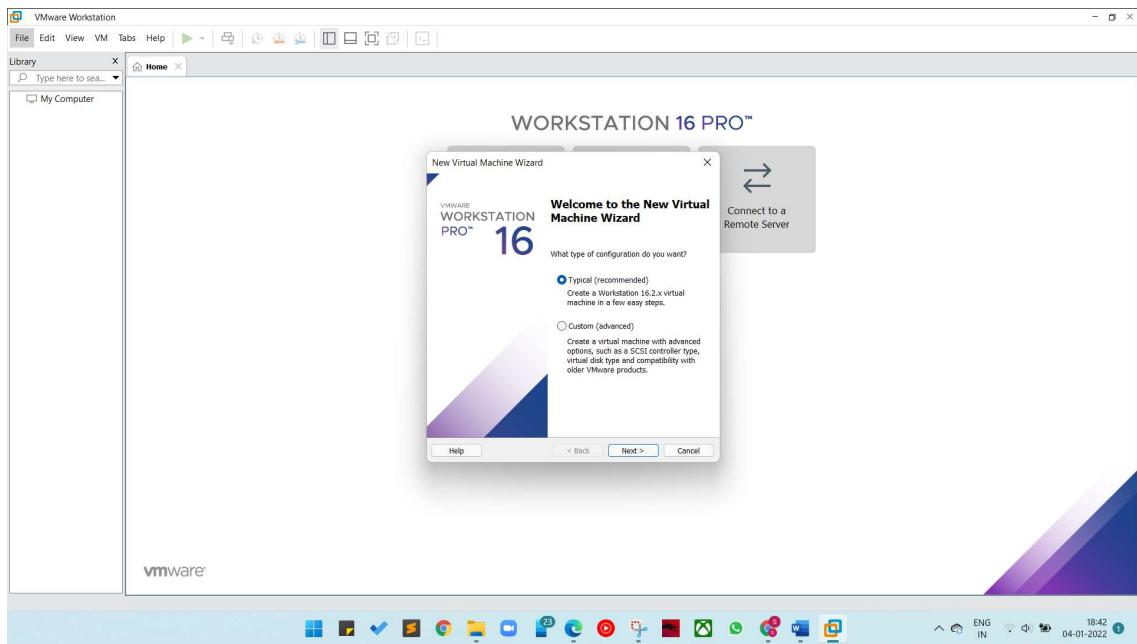
Step 10: For the first time opening, if you have not entered the License key in the previous steps, then it will ask for a license key.



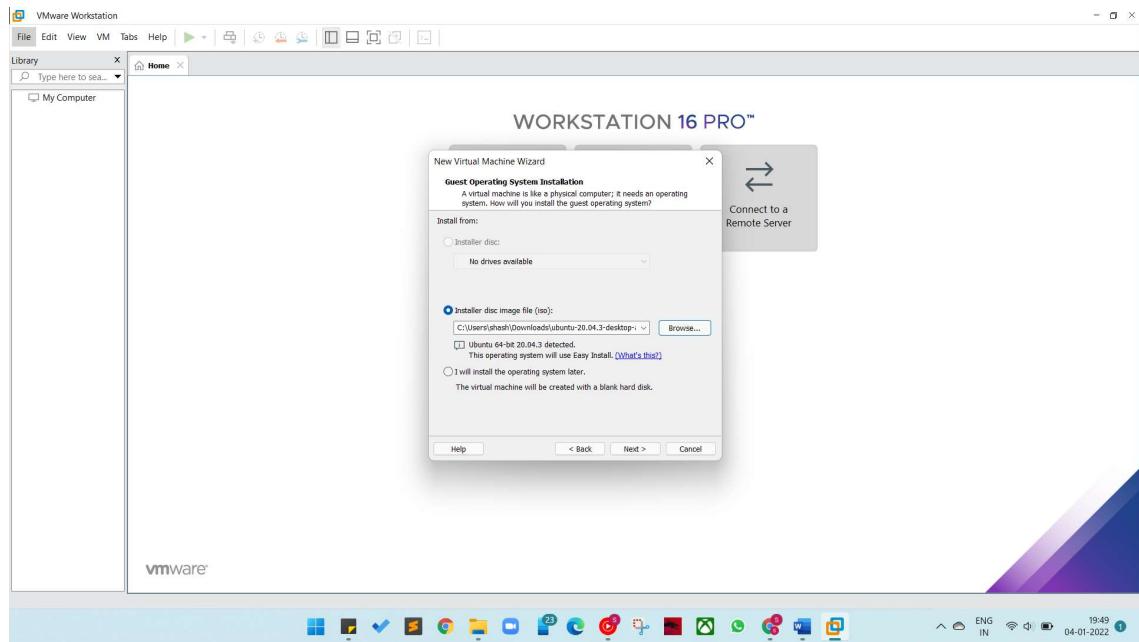
Finally, this will open a window of VMware Workstation Pro.



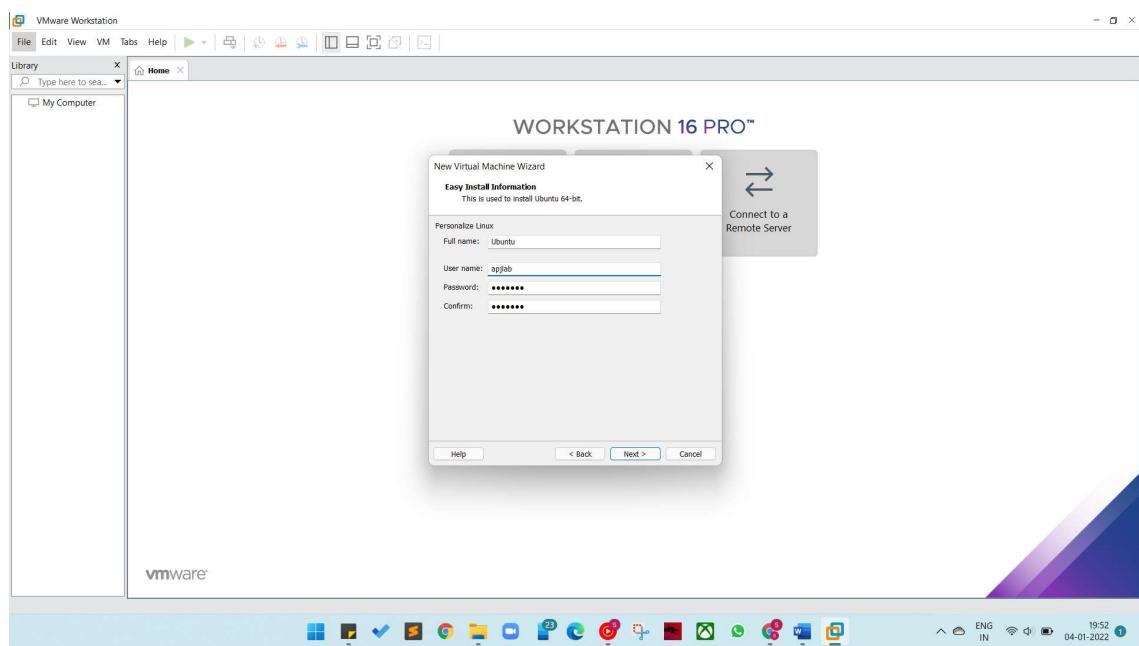
Step 11: Click on File → New Virtual Machine. A New Virtual Machine Wizard will appear. Click on Typical.



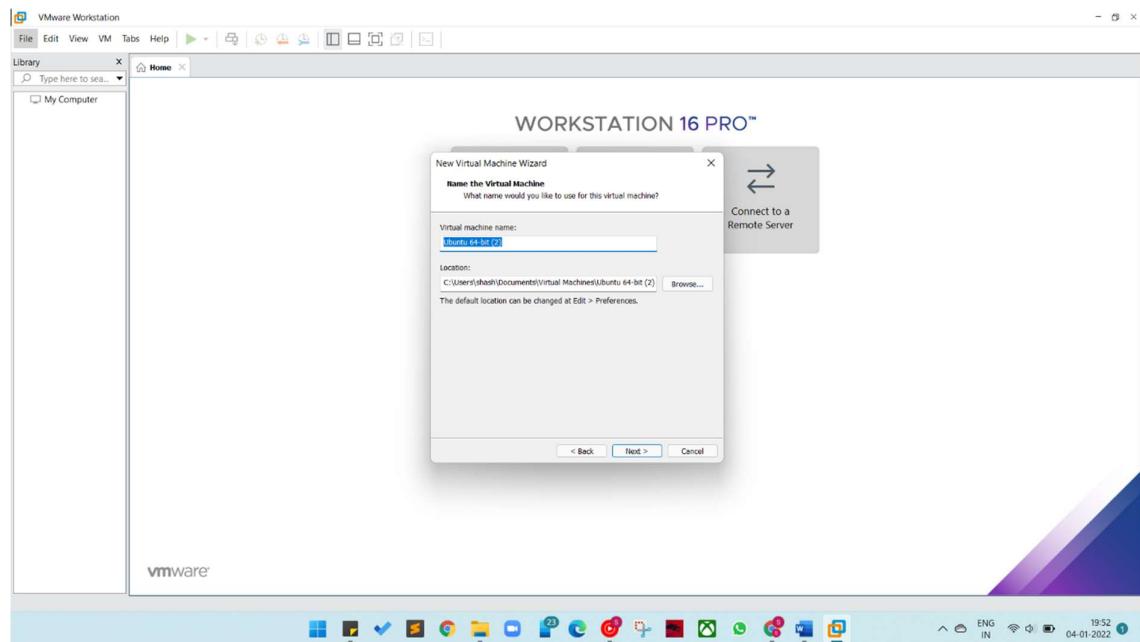
Step 12: Select the ISO File and click on Next



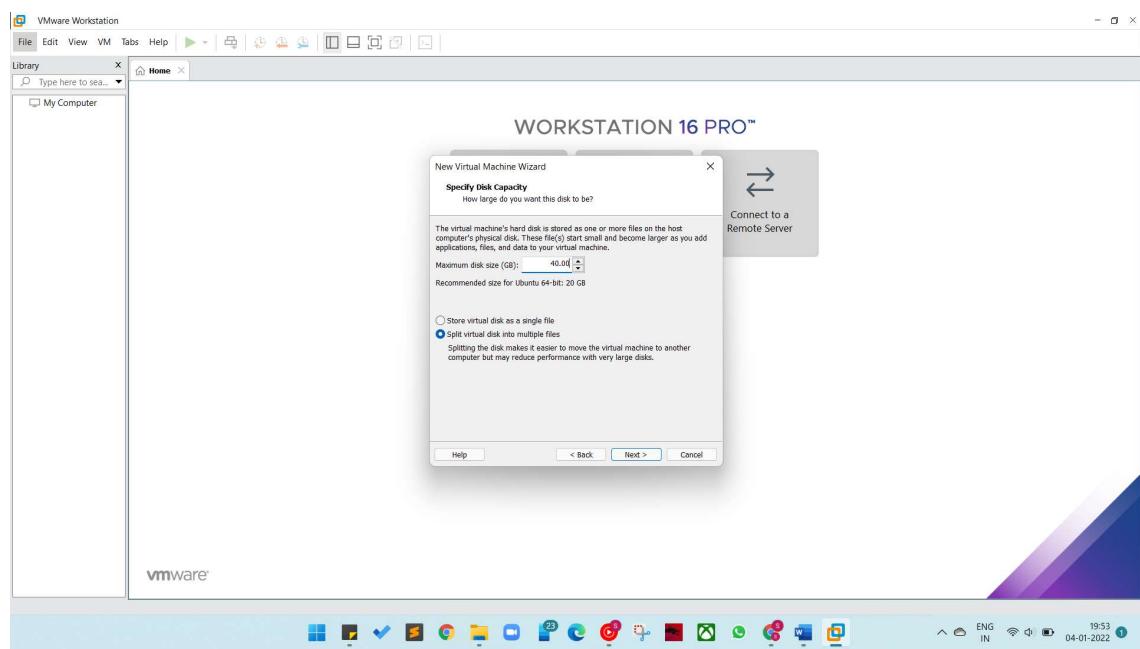
Step 13: Fill in the required details such as username, password and click on next.



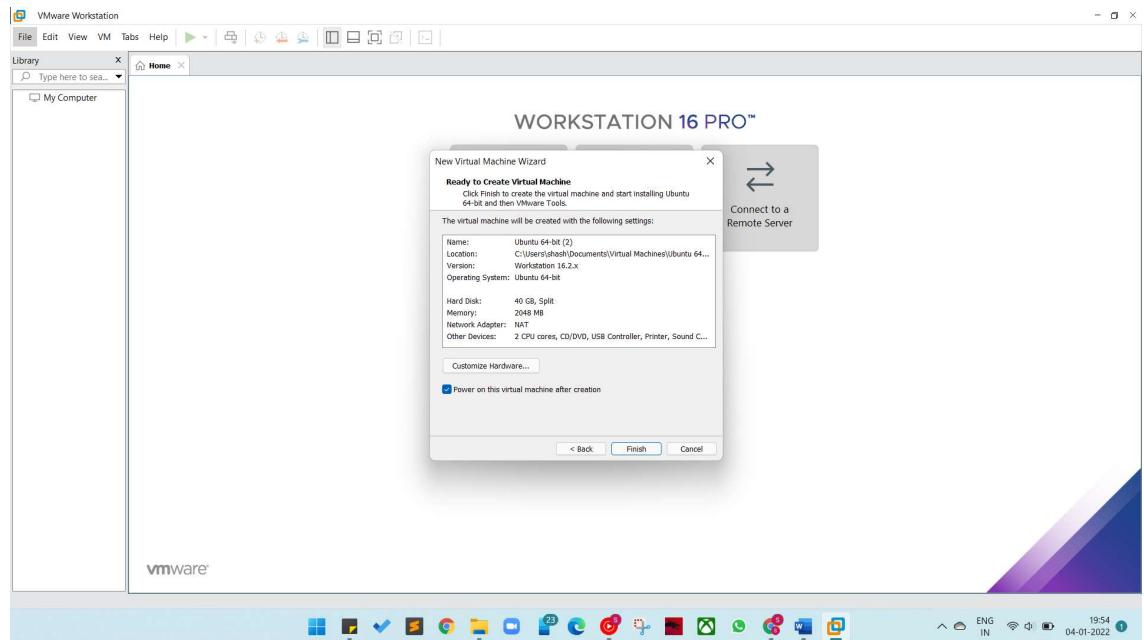
Step 14: Name the virtual machines.



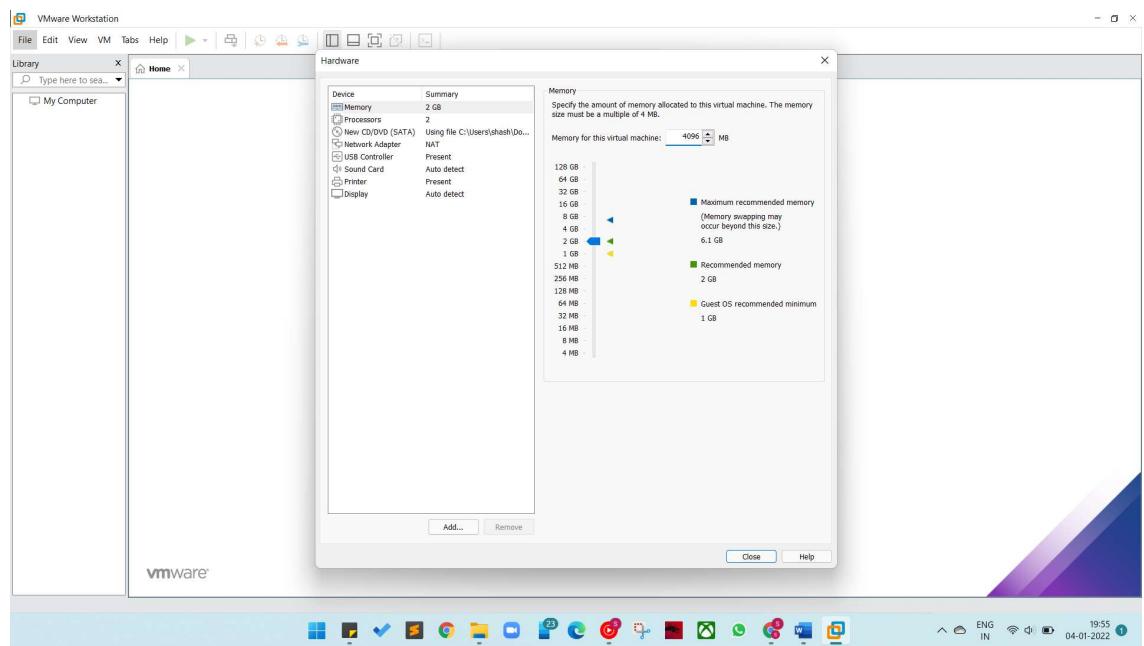
Step 15: Allocate the memory and select Split virtual disk into multiple files and click on next.

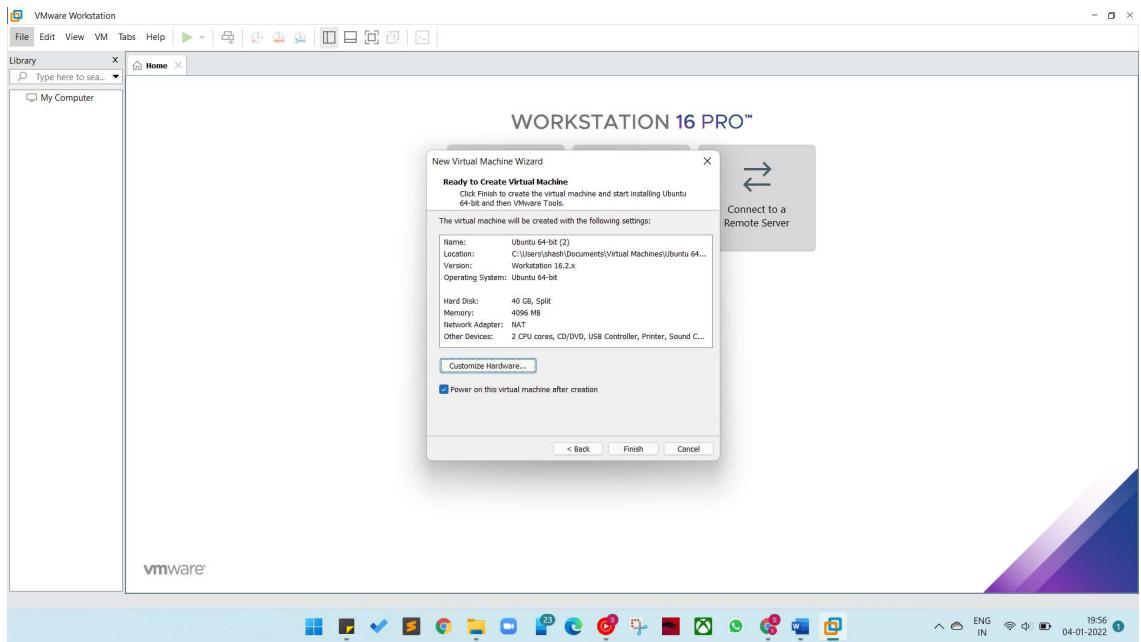


Click on Customize Hardware.

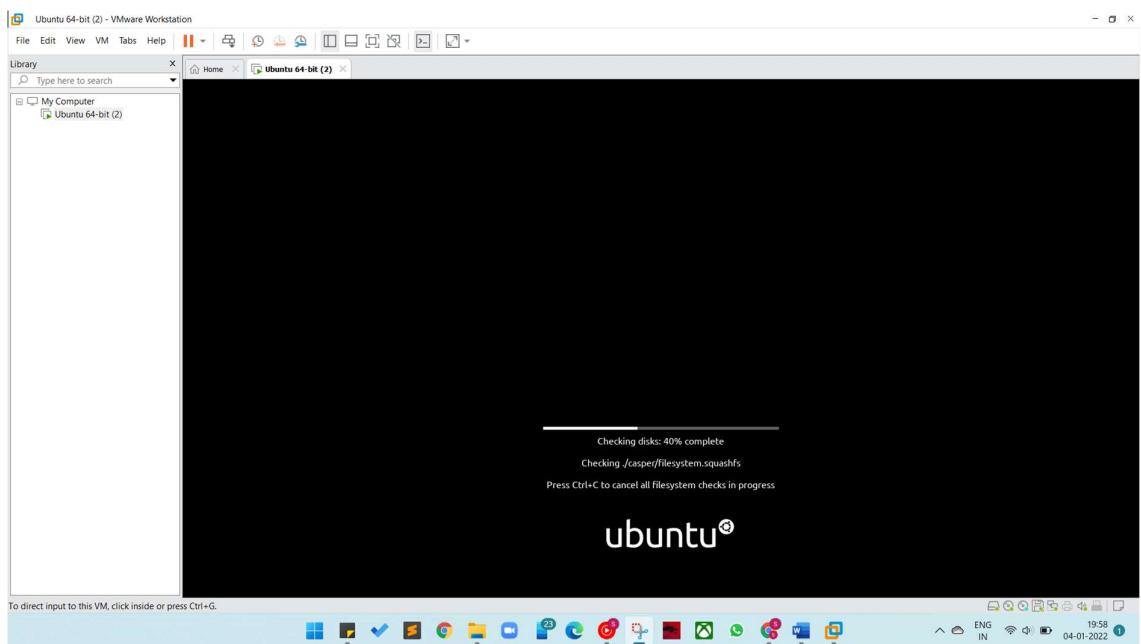


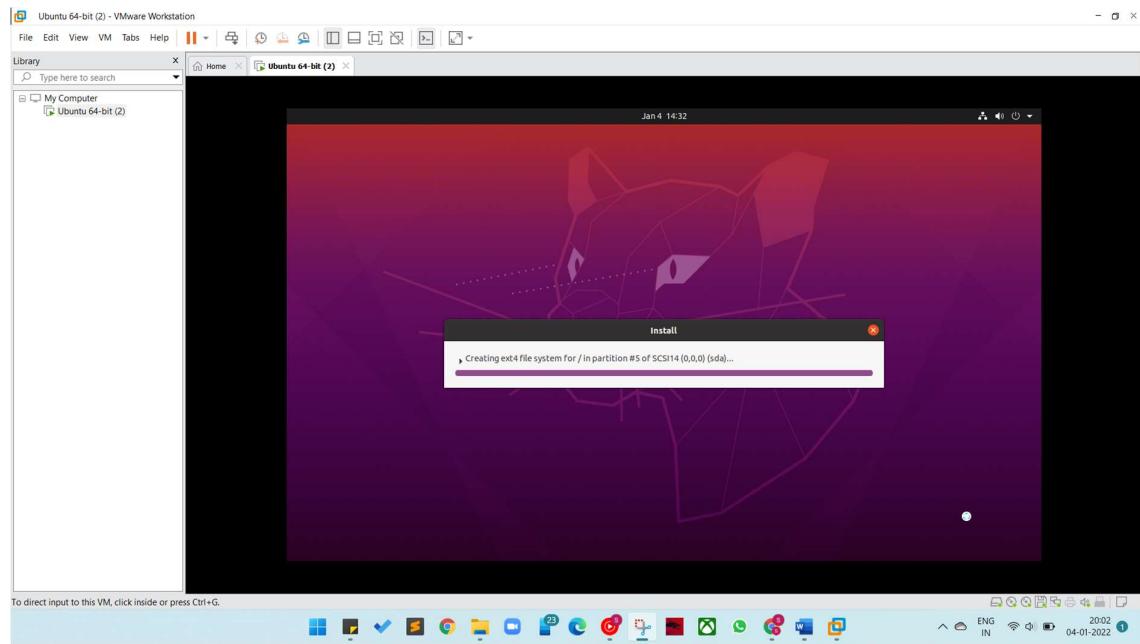
Set the memory size to 4GB



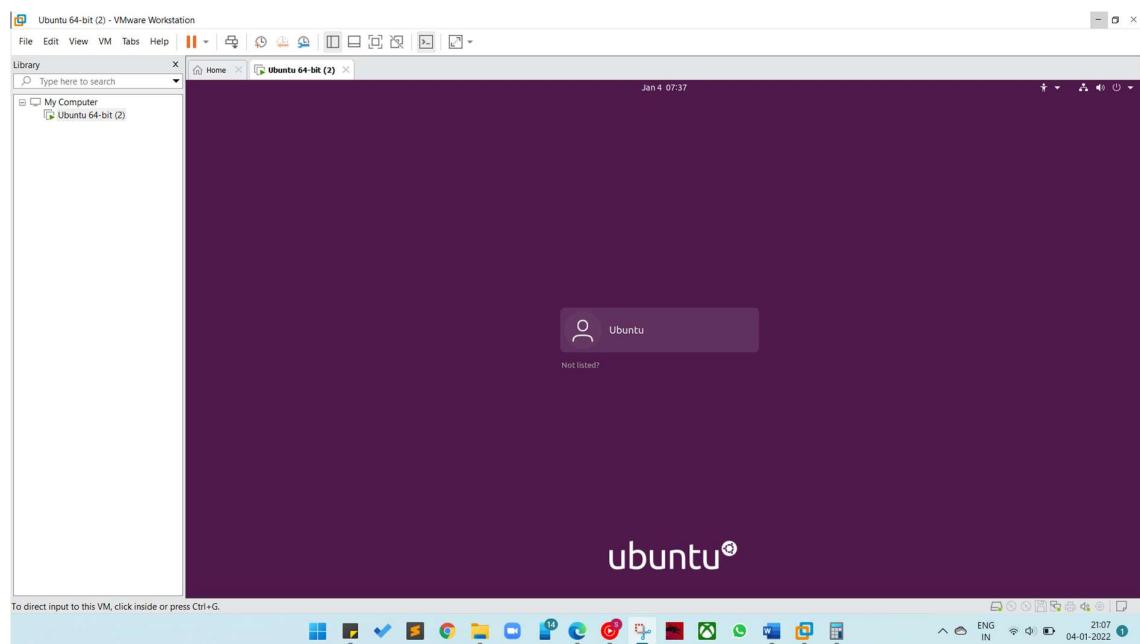


Step 16: Click on Finish. You can see that Ubuntu gets installed in the VM Ware Workstation.



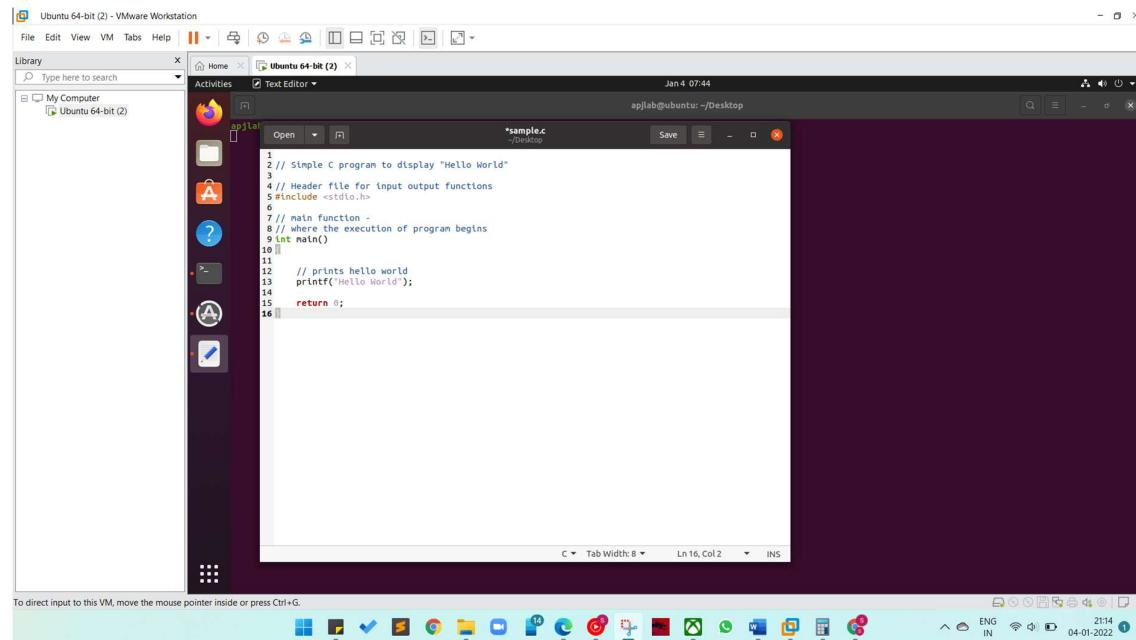
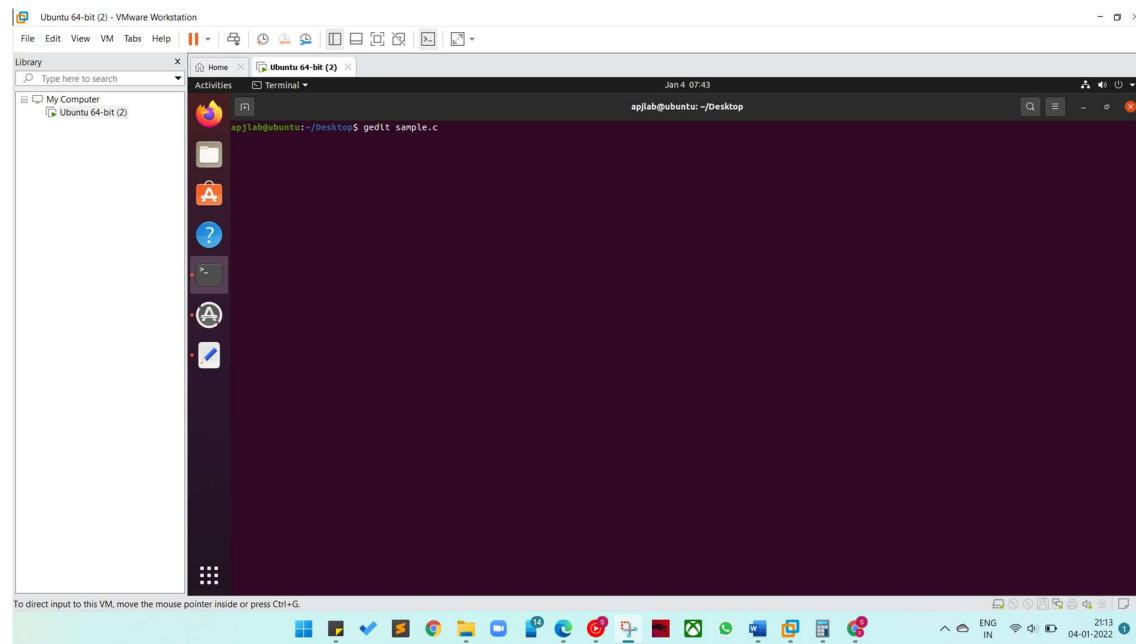


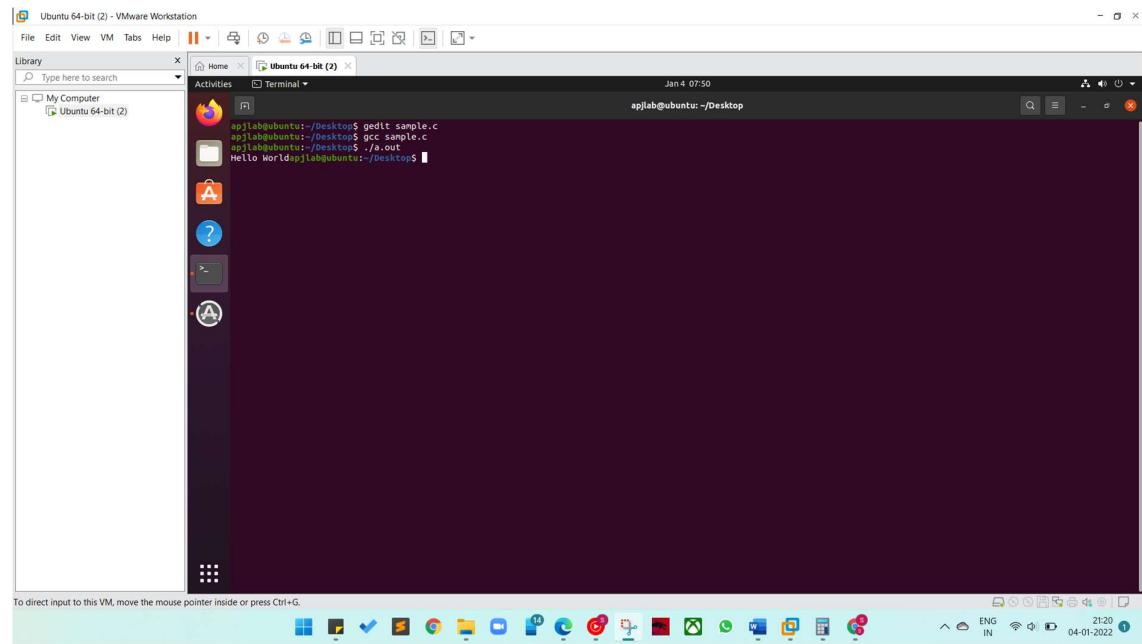
Step 17: Login and Open Terminal and Execute a simple C Program



Step 18: Using gedit write a simple C Code. Use the following commands on the terminal:

- To Open the Editor → **gedit sample.c**
- To Compile the code → **gcc sample.c**
- To Get the output → **./a.out**





VIRTUALIZATION AND FILE SHARING

Virtualization

Virtualization is the process of running a virtual instance of a computer system in a layer abstracted from the actual hardware. Most commonly, it refers to running multiple operating systems on a computer system simultaneously. To the applications running on top of the virtualized machine, it can appear as if they are on their own dedicated machine, where the operating system, libraries, and other programs are unique to the guest virtualized system and unconnected to the host operating system which sits below it.

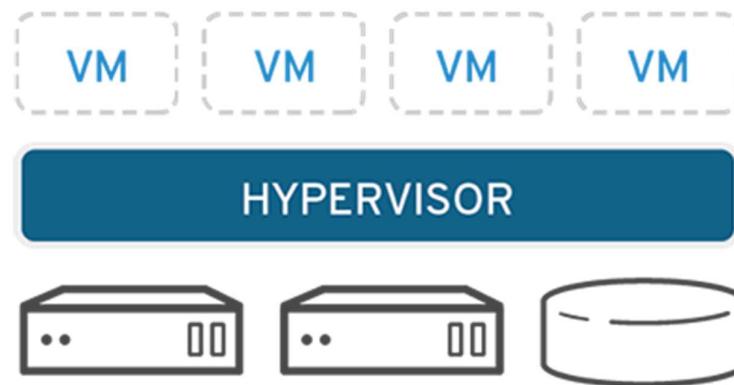
There are many reasons why people utilize virtualization in computing. To desktop users, the most common use is to be able to run applications meant for a different operating system without having to switch computers or reboot into a different system. For administrators of servers, virtualization also offers the ability to run different operating systems, but perhaps, more importantly, it offers a way to segment a large system into many smaller parts, allowing the server to be used more efficiently by a number of different users or applications with different needs. It also allows for isolation, keeping programs running inside of a virtual machine safe from the processes taking place in another virtual machine on the same host.

Working of Virtualization:

Software called hypervisors separate the physical resources from the virtual environments—the things that need those resources. Hypervisors can sit on top of an operating system (like on a laptop) or be installed directly onto hardware (like a server), which is how most enterprises virtualize. Hypervisors take your physical resources and divide them up so that virtual environments can use them.

Resources are partitioned as needed from the physical environment to the many virtual environments. Users interact with and run computations within the virtual environment (typically called a guest machine or virtual machine). The virtual machine functions as a single data file. And like any digital file, it can be moved from one computer to another, opened in either one, and be expected to work the same.

When the virtual environment is running and a user or program issues an instruction that requires additional resources from the physical environment, the hypervisor relays the request to the physical system and caches the changes—which all happens at close to native speed (particularly if the request is sent through an open-source hypervisor based on KVM, the Kernel-based Virtual Machine).



FILE SHARING

Transferring Files to and from Virtual Machines can be done in the following ways:

- Creating a Shared Folder in VirtualBox
- Dragging and Dropping Files in VirtualBox
- Managing Files with NextCloud

Creating a Shared Folder in VirtualBox

A shared folder is a folder that makes its files available on both the guest machine and the host machine at the same time. Creating a shared folder between the guest and the host allows you to easily manage files that should be present on both machines. The course virtual machines are ready to use shared folders right away, but if you are using the virtual machine on your personal computer, then you will need to specify which folder to use as shared storage.

Shared Folders on SCS Lab Computers using Course VMs: If you are using a course VM on a lab computer, it is likely that a shared folder has already been set up for you. On the desktop of your course VM, you should notice a folder titled Shared Folders. Inside this folder, you will find any folders that have been shared between the course VM and lab computers. You should see two folders that have already been configured for you: Z_DRIVE and Temp. Z_DRIVE gives you access to your Windows Account Z:\ drive. This is storage that is persistent to your SCS account and available as a network drive on the lab computers. Temp gives you access to the folder found at D:\temp on the lab computer. Files stored in this folder are local to the machine, meaning that they can be accessed faster but will delete from the system when you log out. If you are working with data that you will need to use again, use the Z_DRIVE for your shared folder. If you need a faster read/write speed, use the Temp folder, but remember to back up your files, or they will be deleted when you log off the computer.

Shared Folders on Personal Computers

If you are using your own personal machine, you will need to configure VirtualBox to look in the right place for your shared files. First, click on the guest machine you intend to share files with. From there, you can select the guest Settings and navigate to Shared Folders on the left side menu. To create a new shared folder, either click the New Folder icon on the right menu or right-click the empty list of shared folders and click Add Shared Folder. From here, there are six options:

Folder Path: The folder name on the host machine. Click the drop-down menu and navigate to the folder you would like to share.

Folder Name: This is the name of the folder as it will appear on the guest machine.

Read-Only: If you check read-only, the guest machine will be unable to write changes to the folder. This is valuable when you only want to send files to the virtual machine, but do not want to risk having the files modified by the guest.

Auto-Mount: When any external storage is connected to a computer, it must be mounted in order to be used. It is recommended that you turn on auto-mounting unless you are familiar with the process of mounting a drive yourself.

Mount Point: Unless you already know about mount points, leave this blank.

Make Permanent: If you check this, the shared folder will be a permanent machine folder. If it is not checked, the folder will not be shared after a shutdown.

On the course virtual machines, when you load into the desktop, you should see a folder labelled Shared Folders. In there, you will see any folders that are currently mounted and being shared.

Dragging and Dropping Files in VirtualBox

If you only need to transfer a few files quickly, you can simply drag and drop the files in. On the top bar of the running guest machine, click on Devices > Drag and Drop and make sure that Bidirectional is selected. This means that you will be able to drag files from the host to the guest and from the guest to the host. Once bidirectional drag and drop is checked, you should be able to begin dragging and dropping files.

NOTE: Sometimes when dragging files into the course VM, you may not be able to drag them into the file browser directly. If you encounter this issue, you should drag your files onto the Desktop and move the files around from there. You should see the cursor change when it is ready to drop files.

You can also drag files from the guest machine into the host. To do this, simply open the file browser on the host to where you would like to drop the files and drag the files from the virtual machine into the file browser of the host. File transfers should be pretty quick; if the virtual machine seems stuck when transferring, simply cancel the transfer and try again.

Managing Files with NextCloud

On any virtual machine, including VirtualBox, VMWare, or the virtual machines hosted on the SCS OpenStack, you can access the SCS NextCloud services to move files between multiple machines and your SCS Windows Account storage. NextCloud offers you all of your SCS storage in one remote location, similar to how you might use other file hosting services like Dropbox or Google Drive. Before trying to use NextCloud, you should check that you can access the service by logging in here. If you can access the NextCloud services, you can browse the various file storage services available to you:

Linux Home: These are the files from your SCS Linux Account

Windows Home: These are the files from your SCS Windows Account and your lab Z:\ drive.

NextCloud: In addition to the other storage accounts provided to you by the SCS, you can also upload up to 20GB of files directly to NextCloud.

With NextCloud, you can upload your files from any machine with an internet connection and download them onto any other machine with an internet connection. For example, you can move project files off of your virtual machine, onto the NextCloud storage, and then download them on your personal laptop. Alternatively, you can upload files from your personal PC onto the NextCloud storage, place them into the Windows home folder, and access those files from either the lab Z:\ drive or download them on a virtual machine like VirtualBox or OpenStack.

Uploading Files to NextCloud from a Lab Computer

If you would like to upload files from a lab computer, the easiest way to do this is to place the files you would like to transfer into your Z:\ drive. These files will be automatically backup into your NextCloud storage under the Windows home folder. After that, you can move them into the main NextCloud storage or choose to keep them in your Z:\drive.

Uploading Files to NextCloud from a VM or Other PC

If you would like to upload files from either a VM or any other computer, you can log in to the NextCloud service using any of the available interfaces, such as the web interface. Press the “+” icon in the top left of the file browser and select Upload File. From here, you can choose to keep it in the main NextCloud storage, move it into your Windows Account storage (the Windows home folder), or into your Linux Account storage (the Linux Home folder).

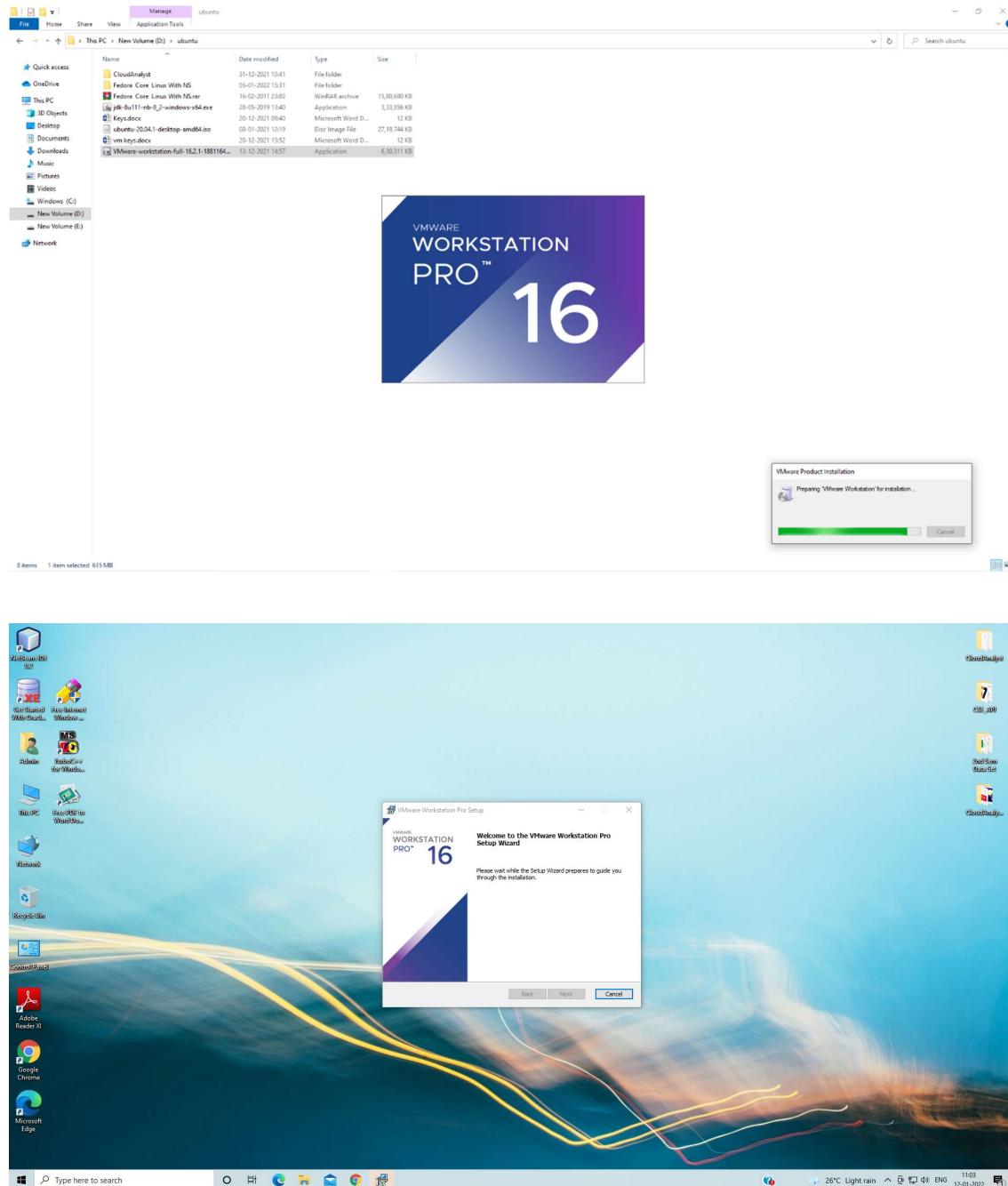
Downloading NextCloud Files to a VM or Other PC

Once your files are uploaded, you will be able to download those files onto any machine, which can connect to NextCloud. First, log in to your preferred NextCloud interface (e.g., the web interface). Navigate to the folder which contains the files you would like to download. Once you are in the target folder, click the checkbox next to each file you would like to download. Above the file listing, you should notice the context bar changing to tell you how many files you have selected and a button labelled Actions. Click Actions > Download.

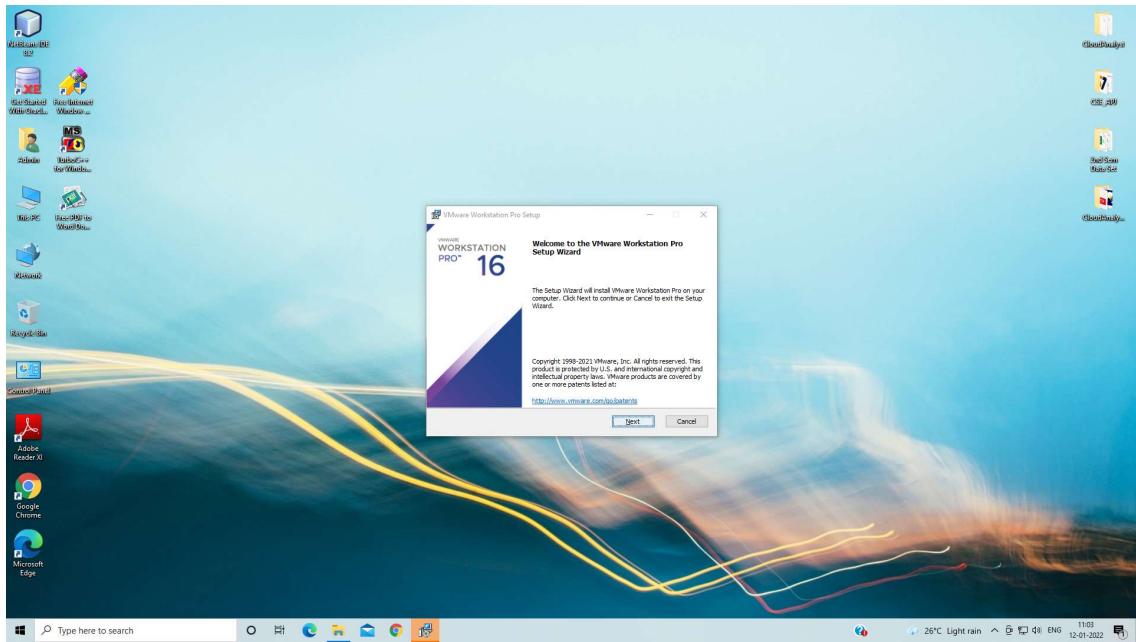
If you selected a single file, it will prompt you to confirm the download. If you have chosen more than one file, NextCloud will place all of the selected files into a zip archive. Before you can use the files, you will need to extract them from the archive. Once you have downloaded your file or extracted your archive, you are ready to use your files on your machine.

2.Create a file in one virtual machine and share it on a host machine.

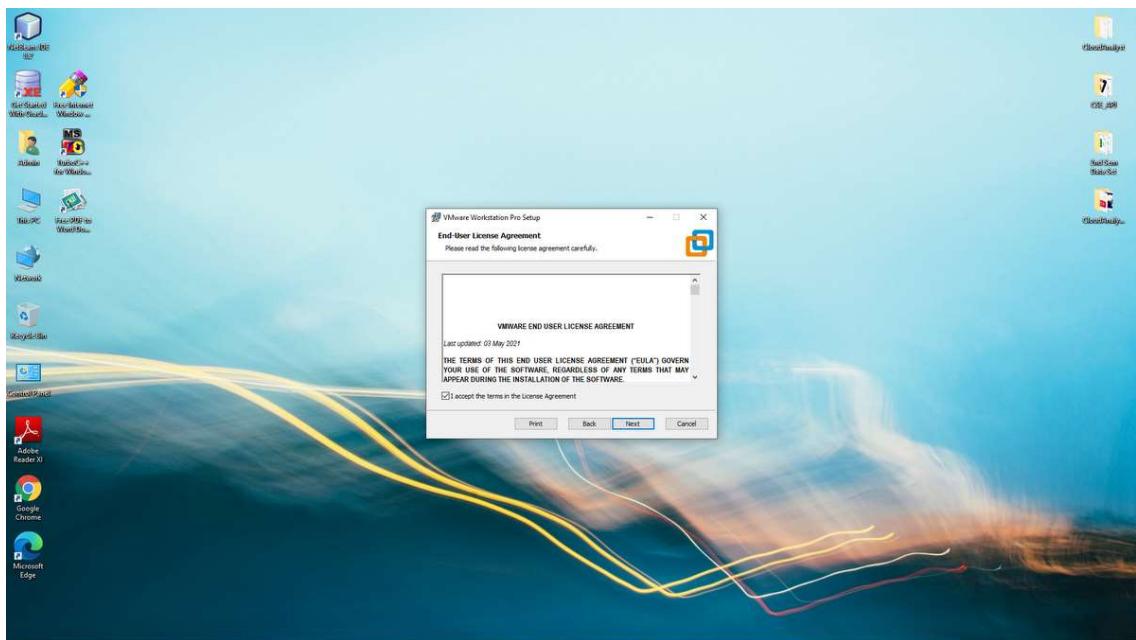
Step 1: Install VM Work Station Pro 16 by clicking on the application file.



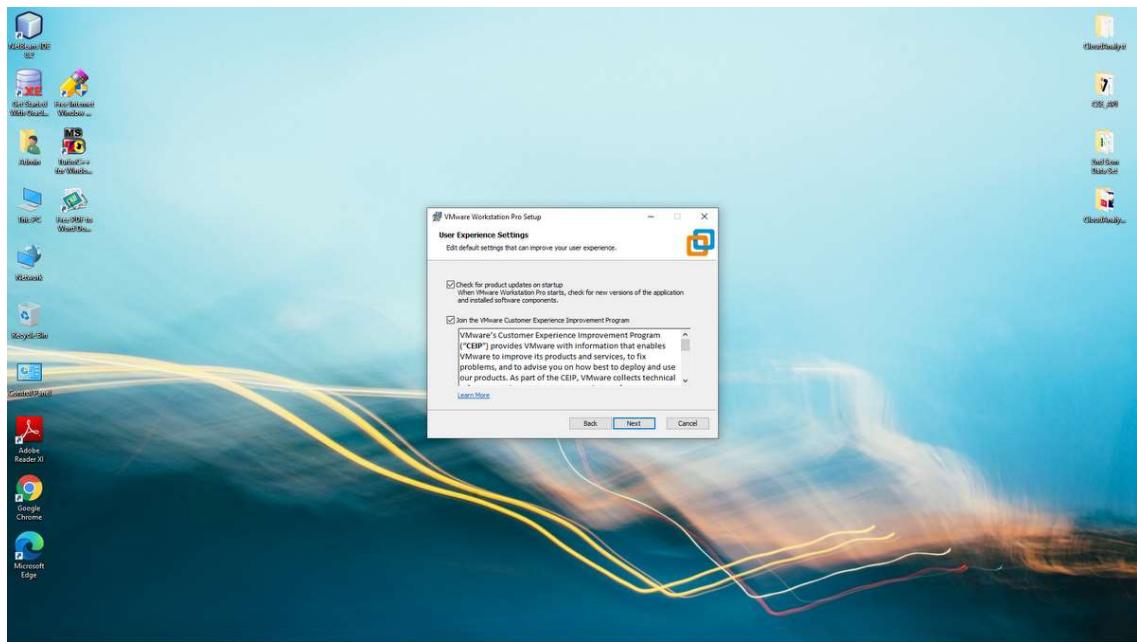
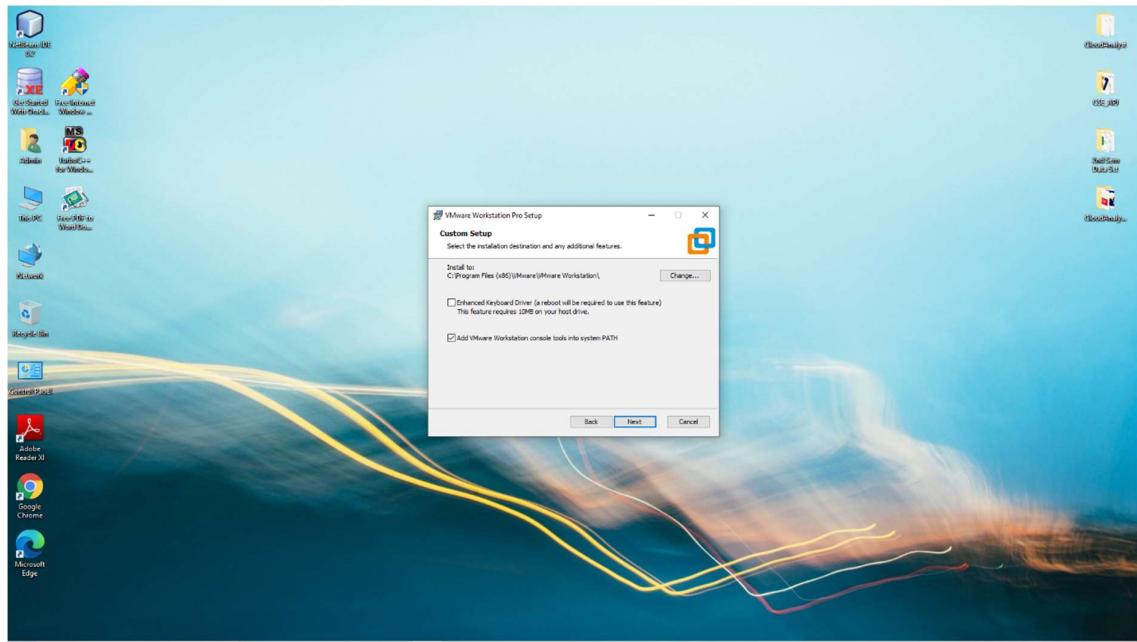
A window appears as shown in the picture. Click on the Next and follow the installation steps.

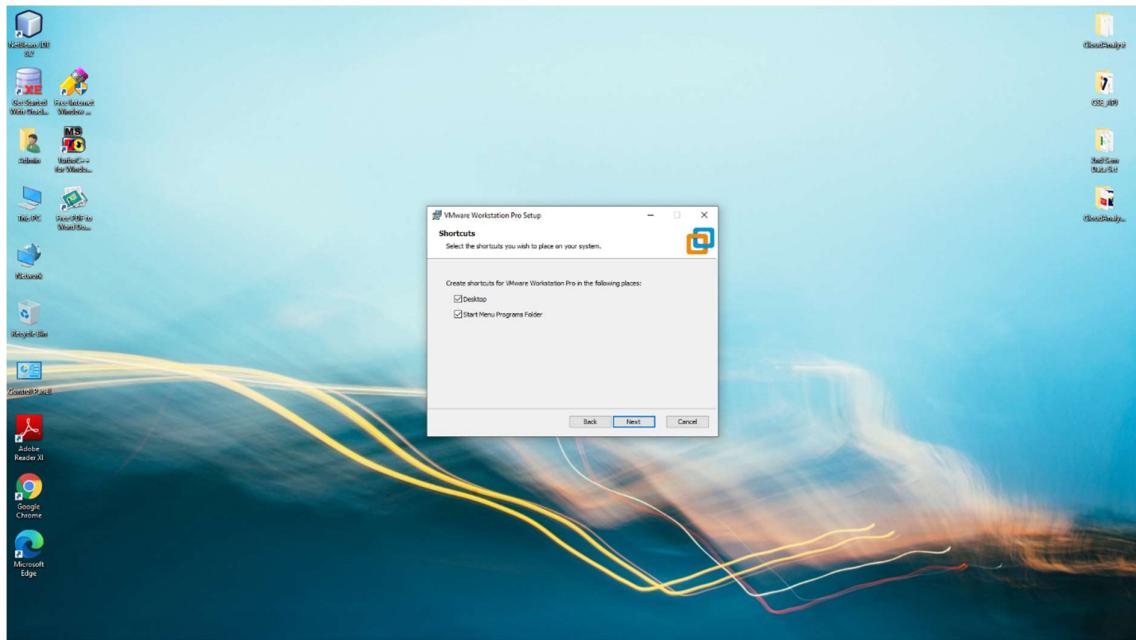


Accept the terms to continue the installation.

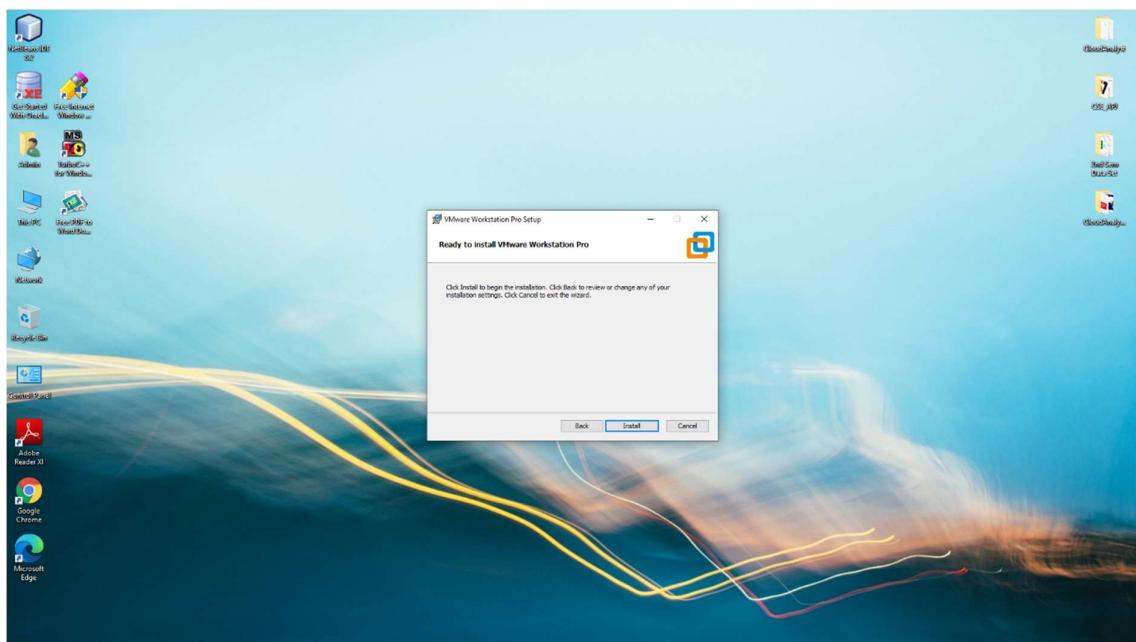


Click on Next.

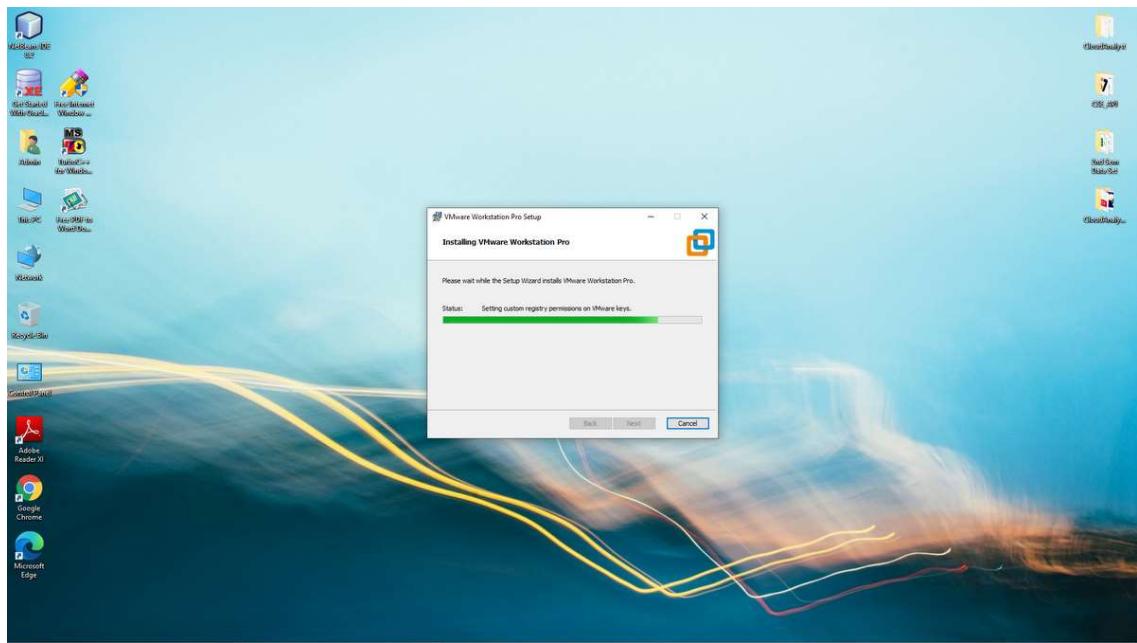




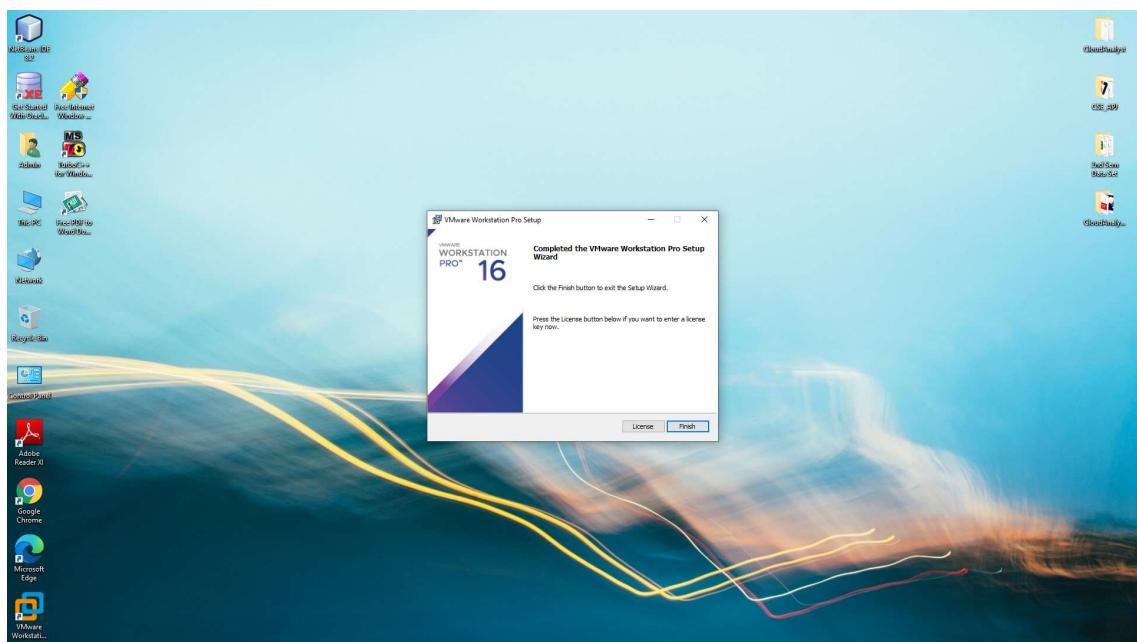
Click on Install



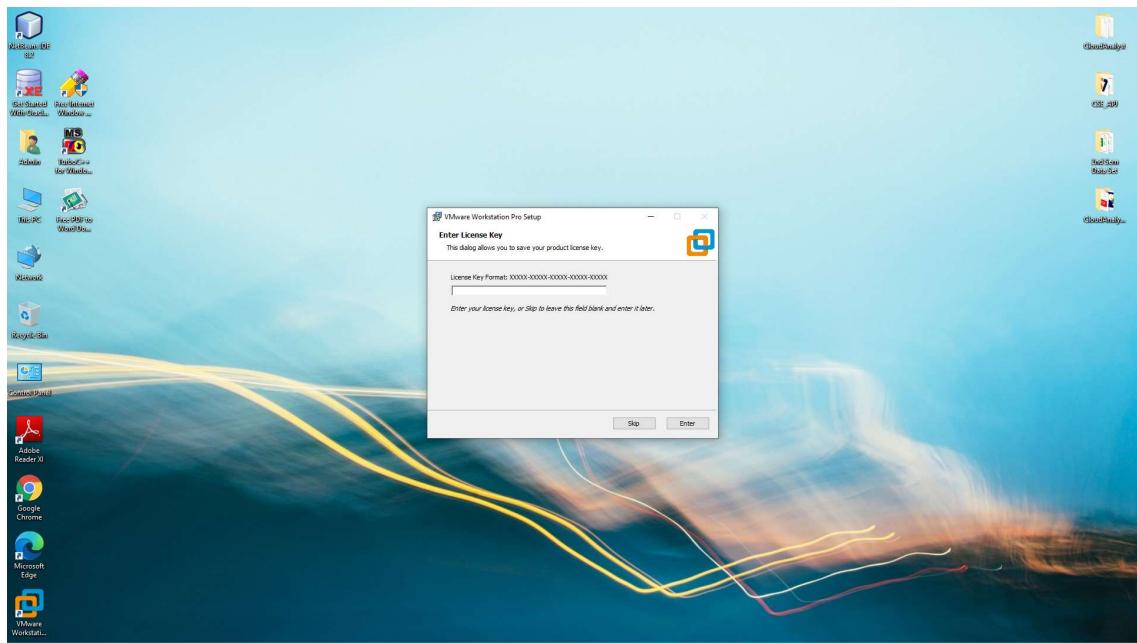
Wait for the installation to complete.



Click on Finish



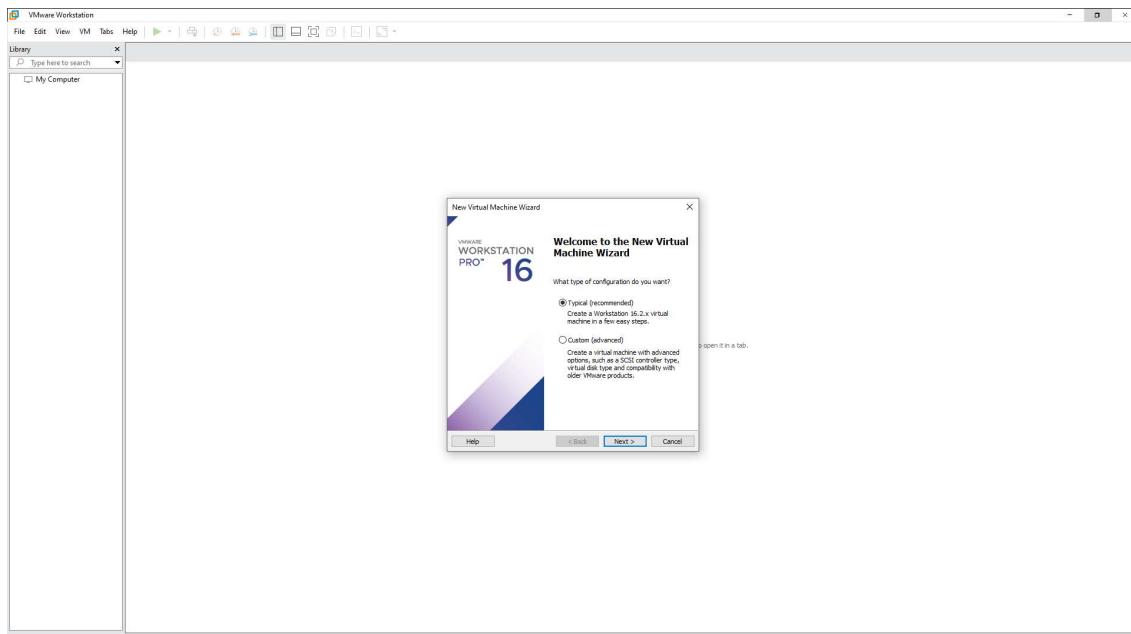
Step 2: Enter the license key received and click on Enter.



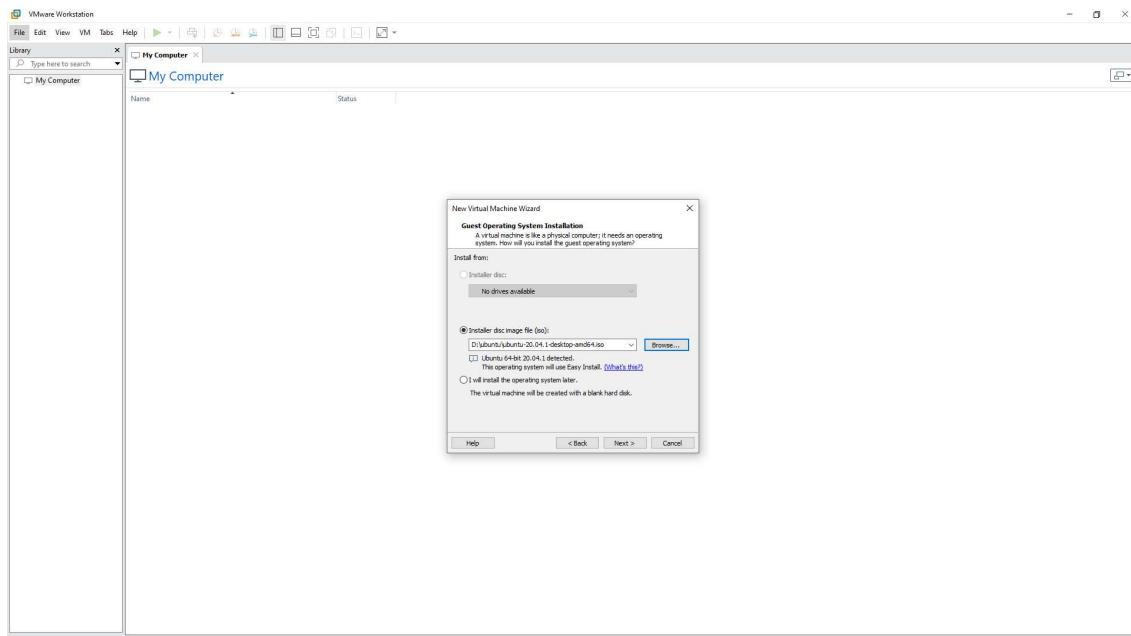
Click on Finish.



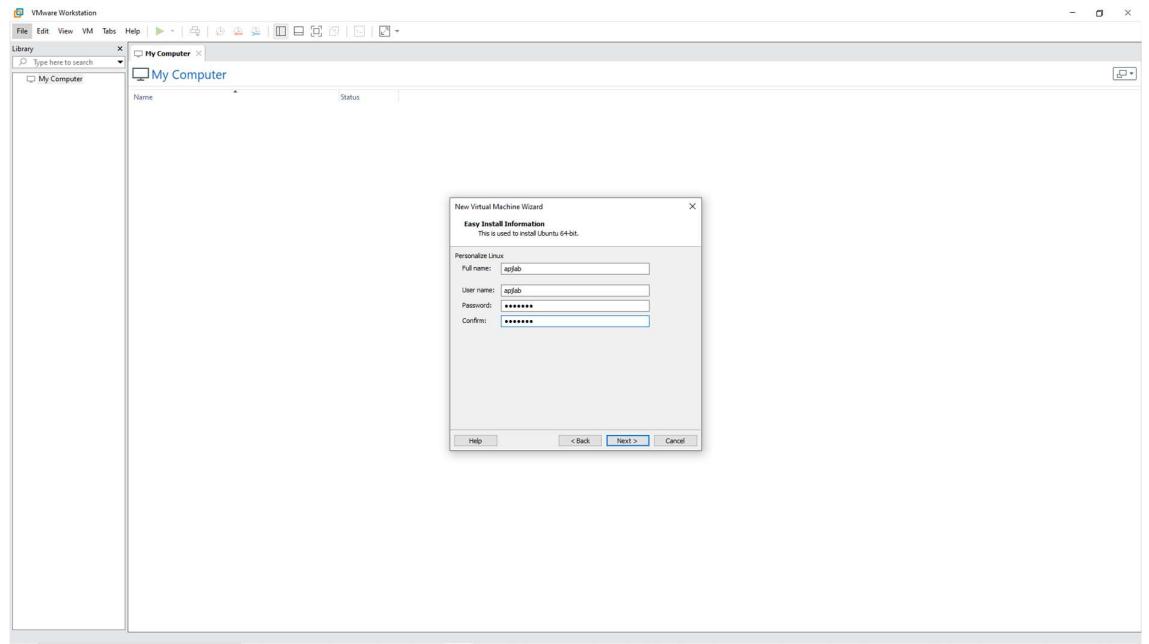
Step 3: Click on File and Click on New Virtual Machine. A window appears as shown in the picture below. Select **Typical** and click on Next.



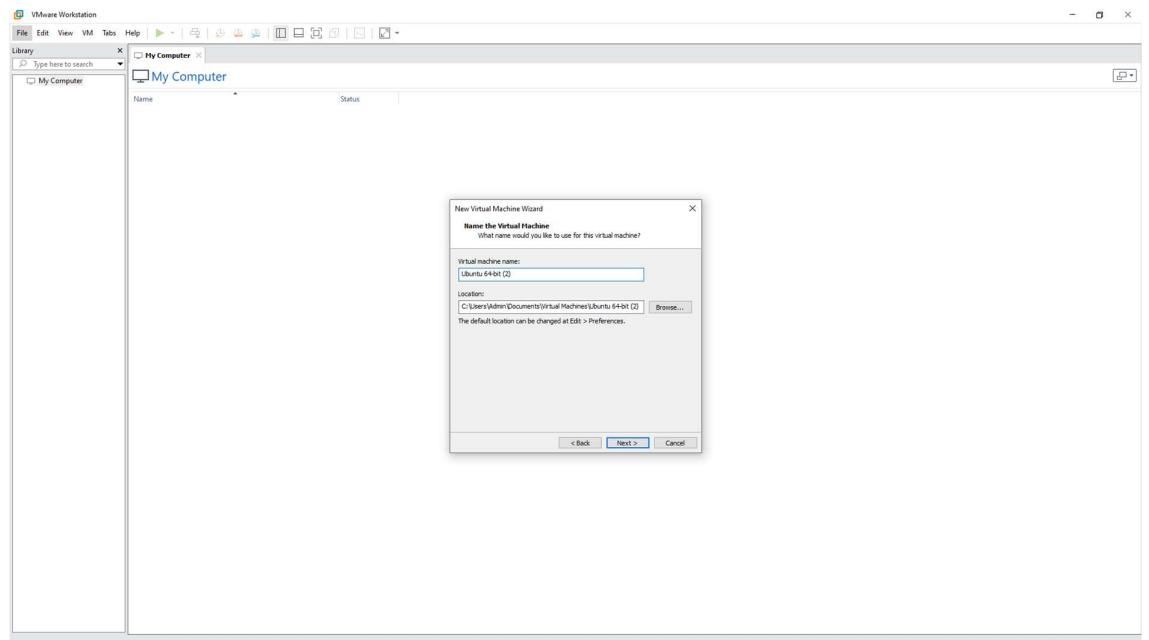
Step 4: Select the Ubuntu ISO file present in the hard disk of your computer. Click on Next.



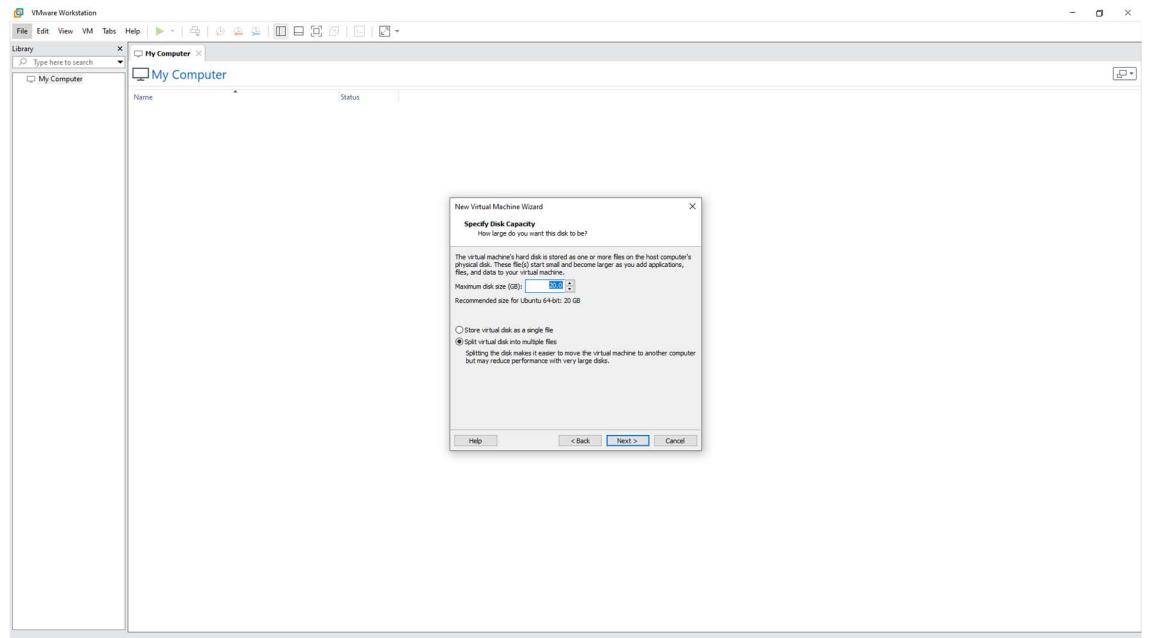
Step 5: Set the Username and Password for your virtual machine and click on next.



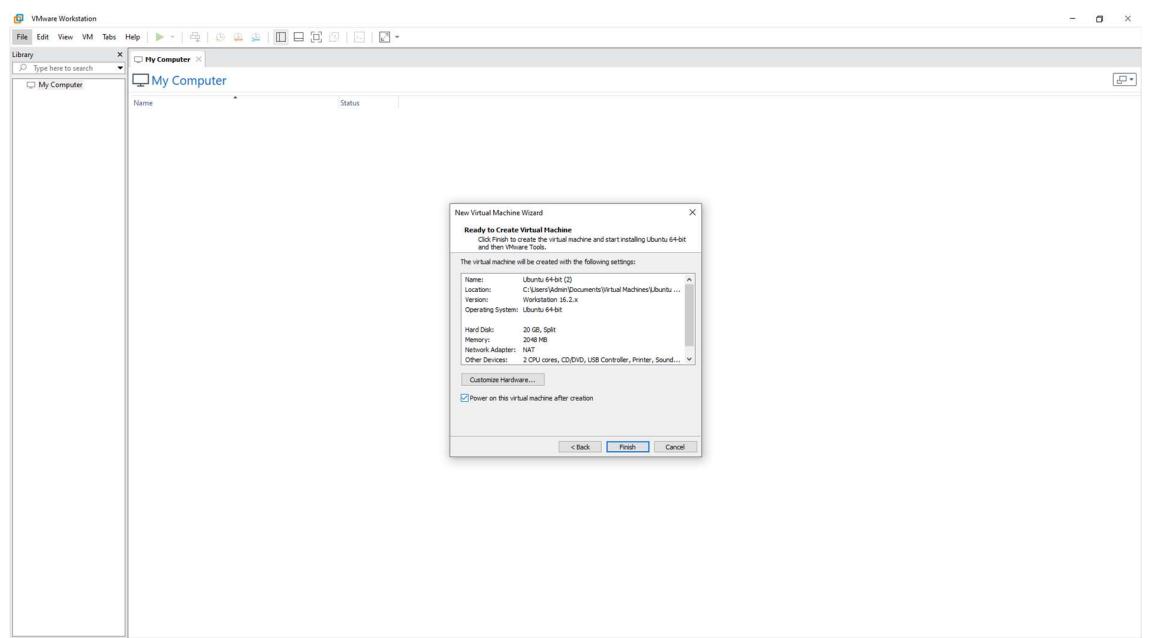
Step 6: Name the virtual machine and click on next.



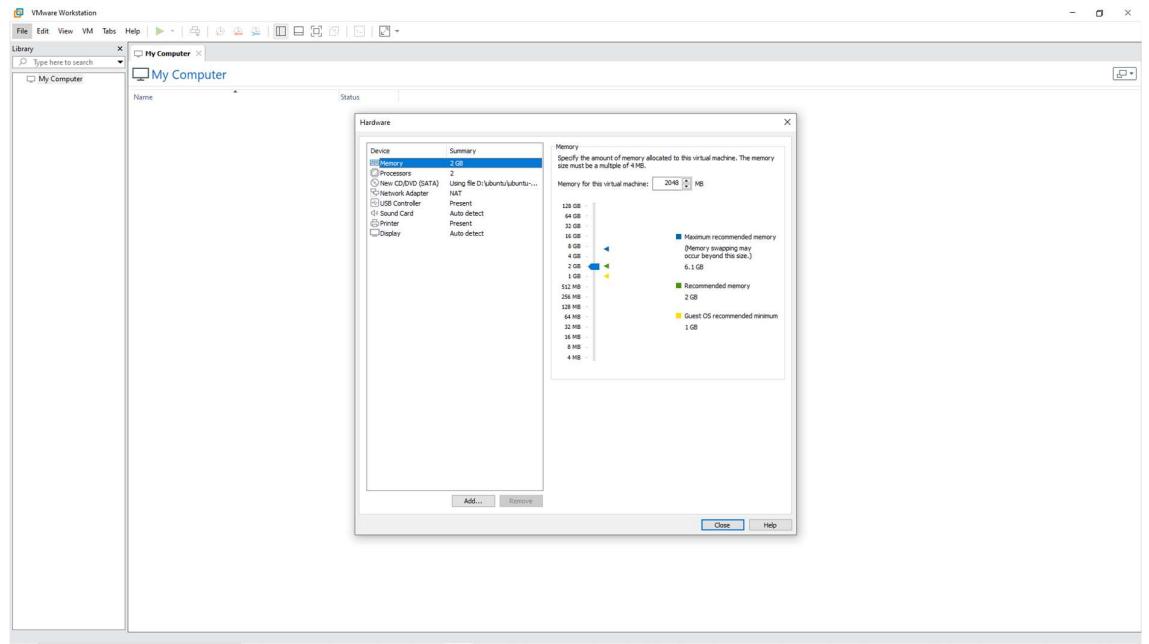
Step 7: Set the disk space to 40GB and click on Next.



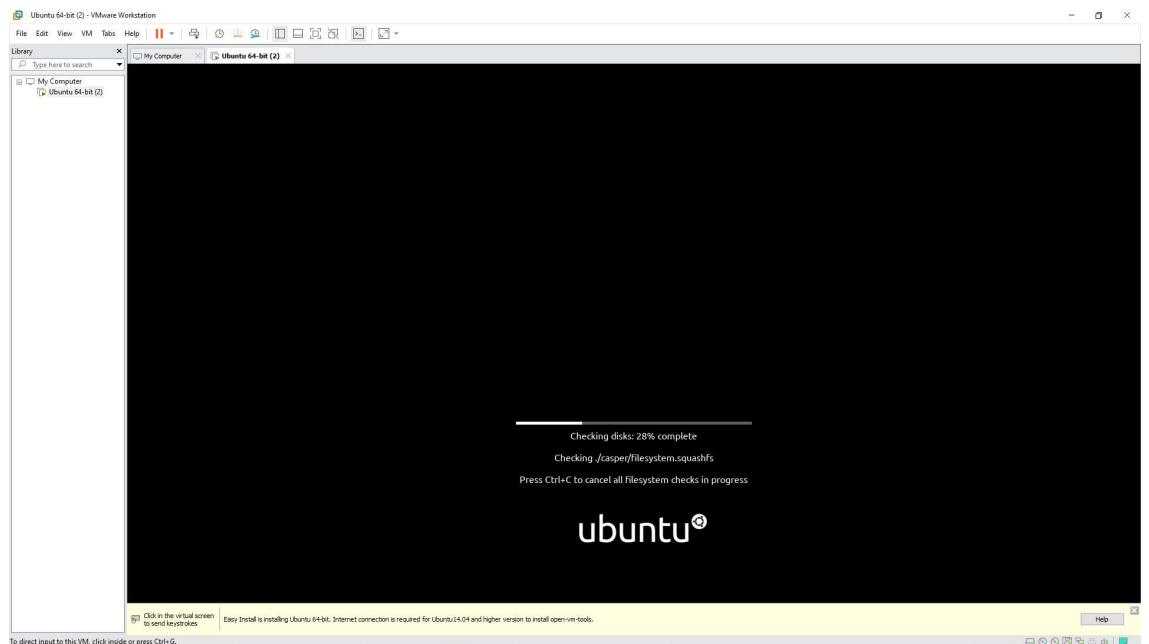
Step 8: Click on Customize Hardware.

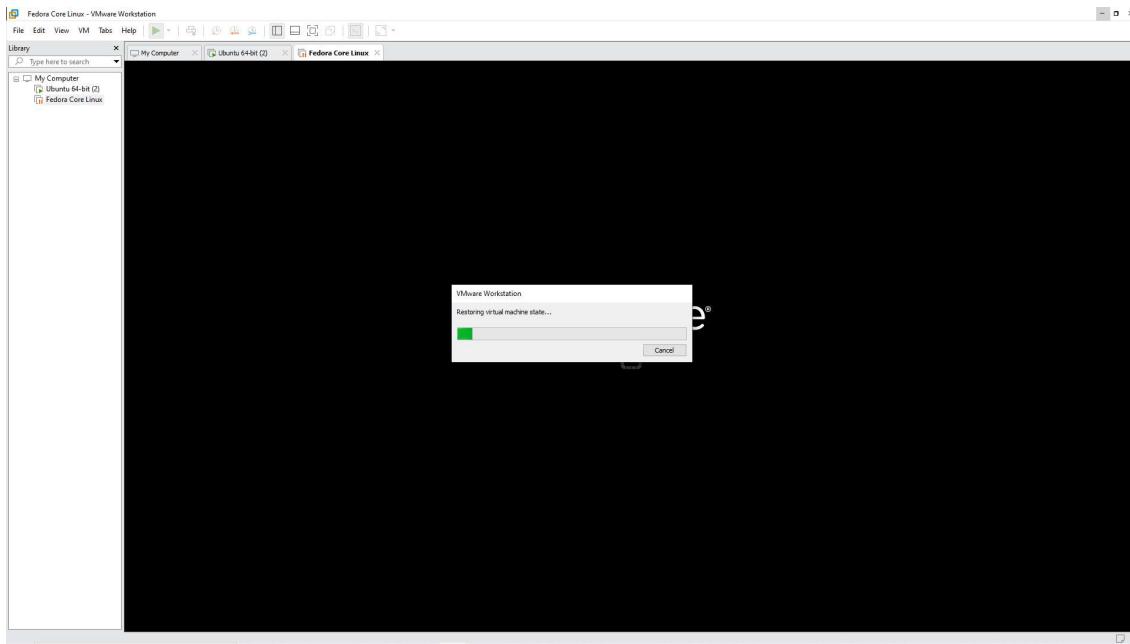


Step 9: Set the RAM Size to 4GB i.e., 4096 MB and click on **Close** and click on **Finish**.



Wait for the installation to complete.

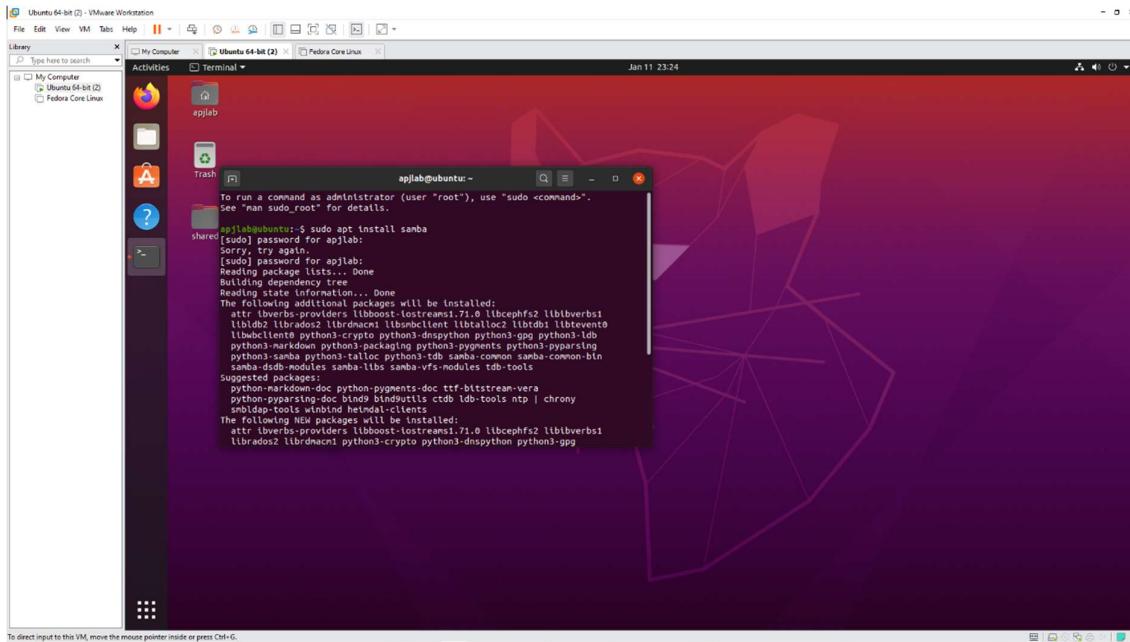




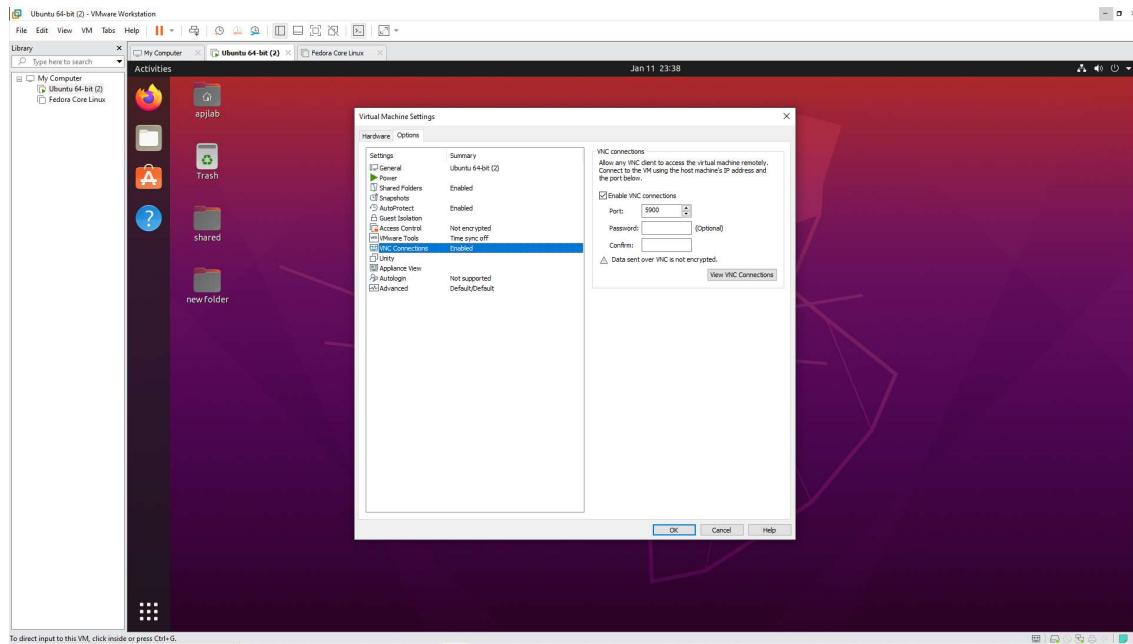
Step 10: Open the terminal and type the following command:

sudo apt install samba

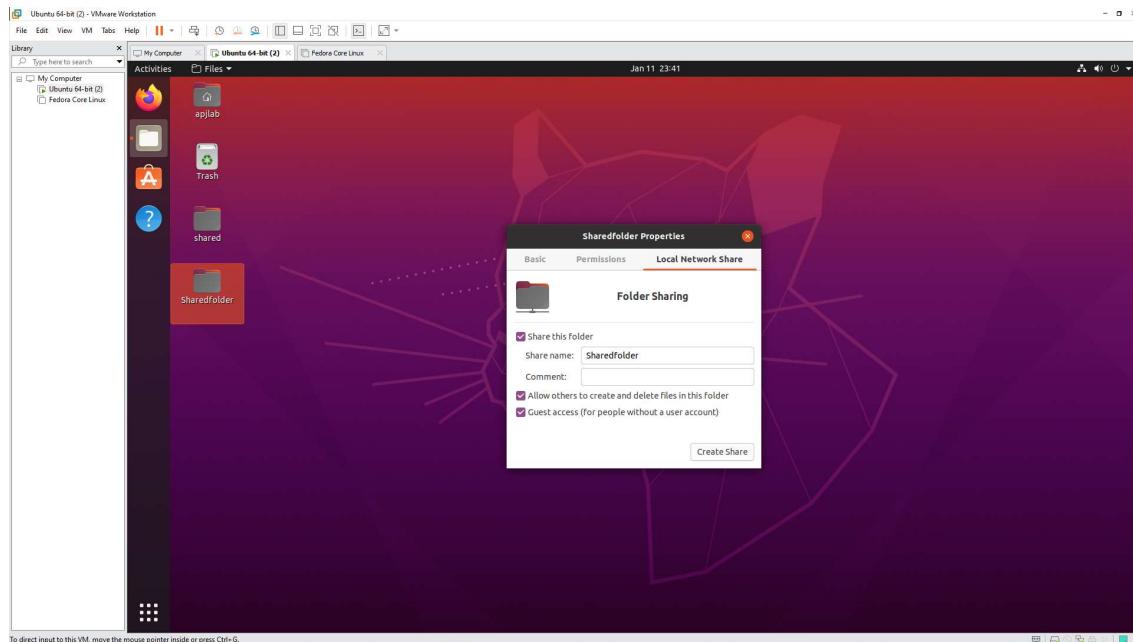
Wait for the installation to complete and close the terminal.



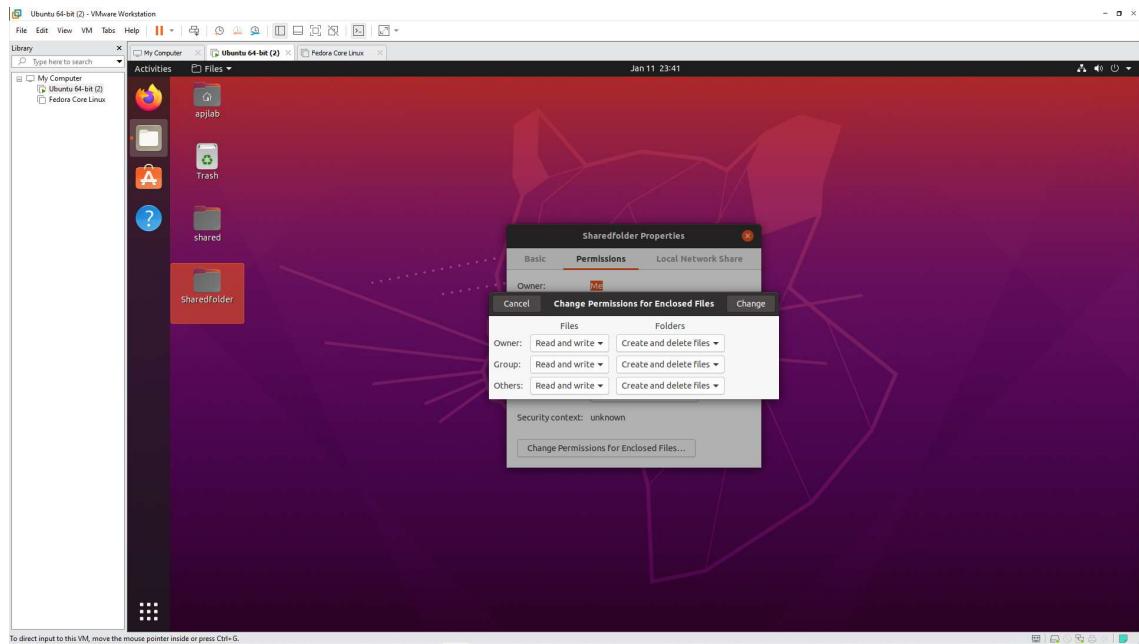
Step 11: Click on VM on top and click on Settings. A window appears. Click on Options and Enable **Shared Folders**, **Auto Protect**, **VNC Connections** and click on **OK**



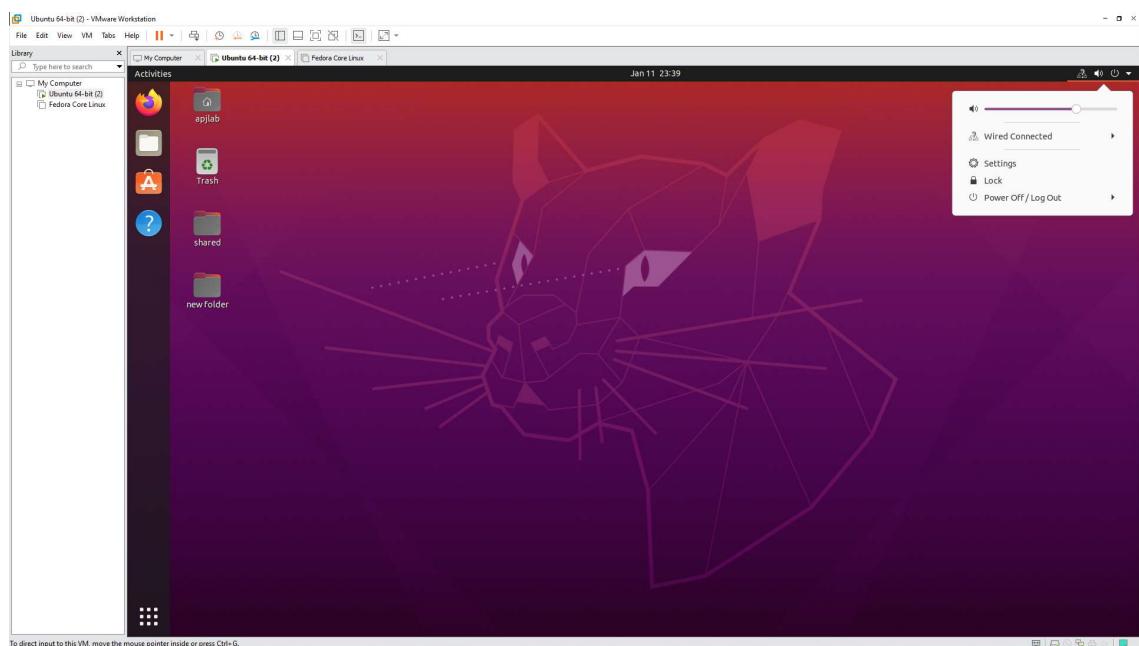
Step 12: Create a new folder that has to be shared and right click on the folder and click on properties. Select **Share this folder, allow others to create and delete files in this folder and Guest access**

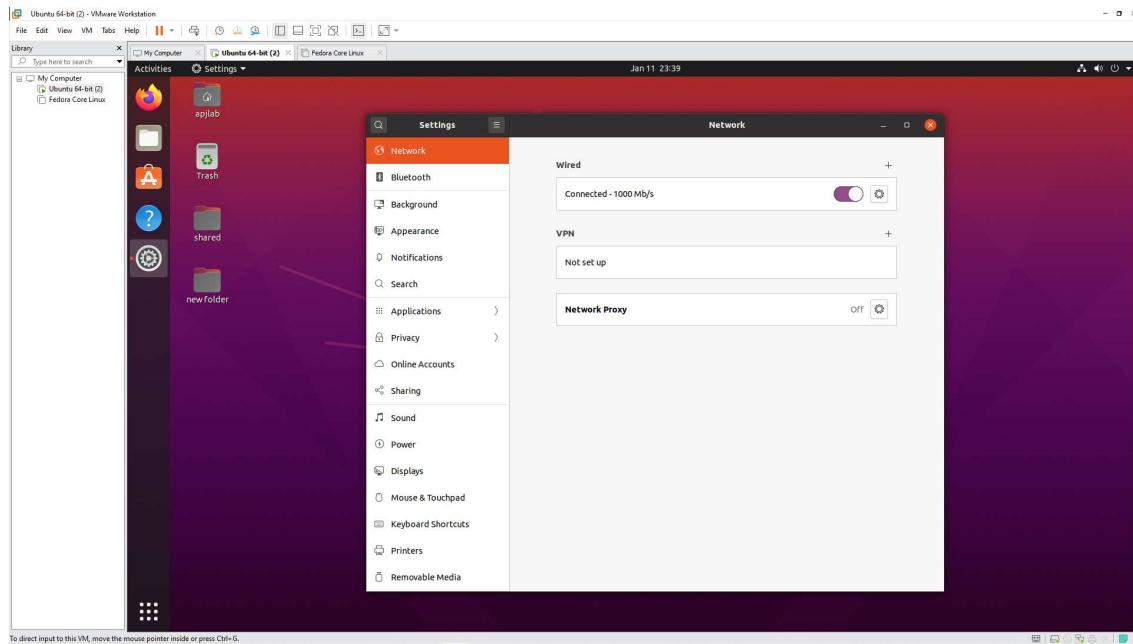


Step 13: Click on permissions and set Read and Write Permissions for Owner, Group and Others.

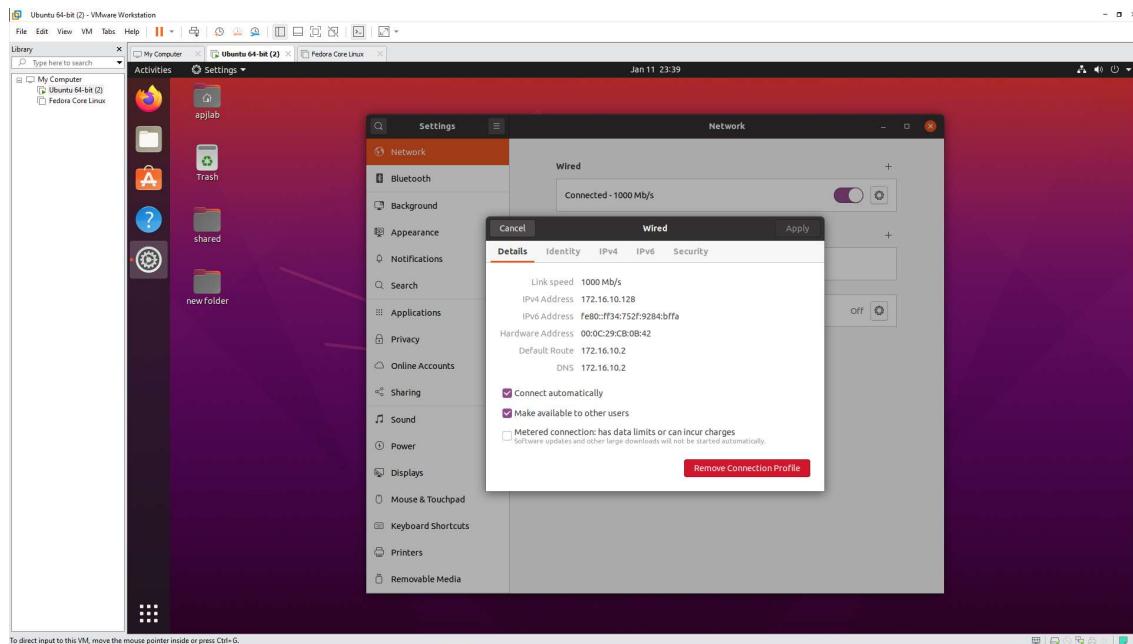


Step 14: Close all the windows and click on Settings

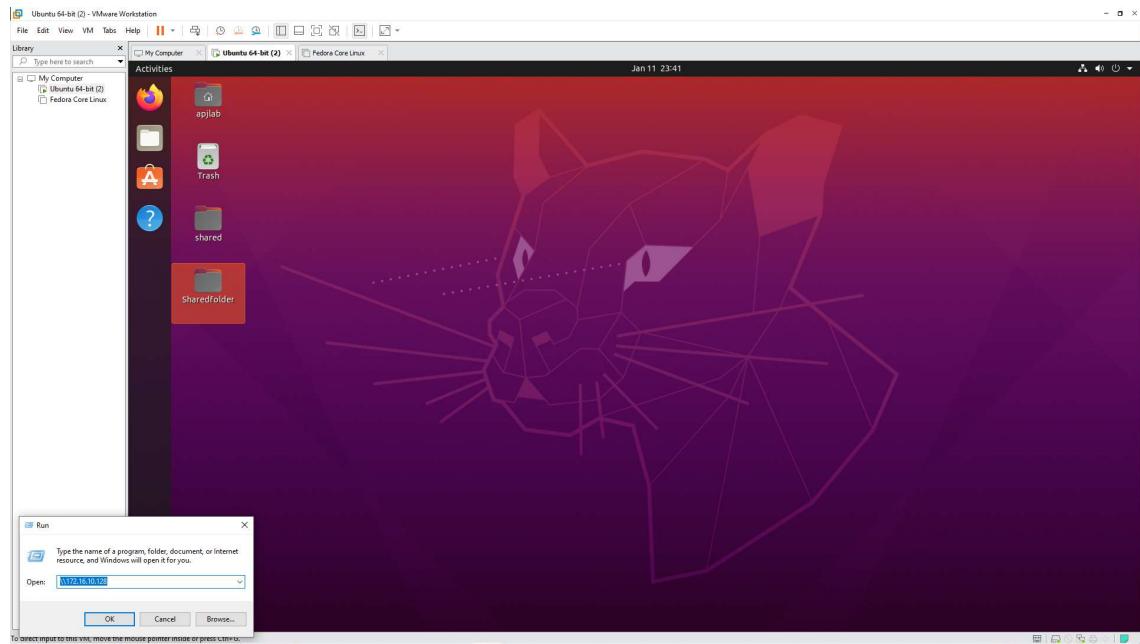




Step 15: Click on Wired Settings and window appears as shown in the below picture. Copy the IPv4 address.



Step 16: Open RUN on Windows and type the IP Address with the “\\”. For example, if the IPv4 Address copied is 172.16.10.128 in the **Run** window enter
\172.16.10.128



Step 17: A window appears showing the folder that was shared. The user can now access the folder.



PART B

Dr. AMBEDKAR INSTITUTE OF TECHNOLOGY

Near Jnana Bharathi Campus, Bengaluru-560 056.

(An Autonomous Institution, Aided by Government of Karnataka)



MINI PROJECT REPORT

ON

“FLIGHT PRICE PREDICTION USING GOOGLE APP ENGINE”

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

SUBMITTED

BY

SHASHANK M KADIWAL

1DA18CS141

UNDER THE GUIDANCE OF

**Mr. Praveena M V
Asst.Prof, Dept. of CSE
Dr.AIT**

Department of Computer Science & Engineering

2021-22

Dr. AMBEDKAR INSTITUTE OF TECHNOLOGY

Near Jnana Bharathi Campus, Bengaluru-560 056.

(An Autonomous Institution, Aided by Government of Karnataka)



CERTIFICATE

This is to certify that the project entitled "**FLIGHT PRICE PREDICTION USING GOOGLE APP ENGINE**" submitted in the partial fulfillment of the requirement of the 7th semester Cloud Computing laboratory curriculum during the year 2021-22 is a result of bonafide work carried out by-

**SHASHANK M KADIWAL
1DA18CS141**

Signature of the guide

**Mr. Praveena MV
Asst. Prof., Dept. of CSE
Dr. AIT**

1. Internal Examiner _____

2. External Examiner _____

**Dr. Siddaraju,
Head of Department
Department of CSE, Dr.AIT**

ACKNOWLEDGEMENT

The sense of contention and elation that accompanies the success of this seminar and the report could be incomplete without mentioning the names of people who have helped me in accomplishing them, people whose constant guidance, support and encouragement resulted in the realization.

I consider myself privileged to express our gratitude and respect towards all those who guided me through the project, "**FLIGHT PRICE PREDICTION USING GOOGLE APP ENGINE**".

I take this opportunity to thank, **Dr. M Meenakshi, Principal, Dr. Ambedkar Institute of Technology, Bengaluru** for his support and encouragement.

I am grateful to **Dr. Siddaraju, Head of Department, CSE, Dr. Ambedkar Institute of Technology, Bengaluru** for providing encouragement and support.

I consider ourselves privileged to express our gratitude and respect towards our guide **Mr. Praveena M V** for constant guidance and support for the completion of the project.

Lastly, I thank all the members of the staff both teaching and non-teaching, friends and last but not the least our parents and family, for helping me directly or indirectly in the completion of the project.

SHASHANK M KADIWAL

ABSTRACT

The proposed system is an automated system that is used to predict the flight price based on various factors. This was with a view to eliminate the problem of using complex algorithms to calculate the flight prices by the airline companies that take into consideration of various factors such as financial, marketing and various other social factors. This also avoids the difficulty of the airlines to maintain the prices to change dynamically based on different conditions. The system also helps the customer to predict future flight prices and plan their journey accordingly.

The proposed system was tested using the information collected from data by various airlines and compared with the existing system. The design provides excellent flight price prediction service and improved information structure.

CONTENTS

<u>Chapter No.</u>	<u>Title</u>	<u>Page No.</u>
Chapter 1	Introduction	181
Chapter 2	Literature Review	182
2.1	Python	182
2.2	Google App Engine	184
2.3	Jupyter Notebook	185
2.4	HTML	185
2.5	CSS	185
2.6	Flask	185
2.7	Numpy	185
2.8	Pandas	186
2.9	Scikit Learn	186
Chapter 3	Requirement Specification	187
3.1	Hardware Requirements	187
3.2	Software Requirements	187
Chapter 4	Design	188
4.1	Data Gathering and Preparation	189
4.2	Data Wrangling	189
4.3	Data Analysis	190
4.4	Train Model	190
4.5	Test Model	192
4.6	Deployment	193
Chapter 5	Development	194
Chapter 6	Results	216
Chapter 7	Deployment	218
7.1	Cloud Deployment	218
7.2	Deploying on Google App Engine	220
7.3	Results and Dashboard	222
	Conclusion	
	Bibliography	

CHAPTER 1

INTRODUCTION

A Flight Price Prediction System using Google App Engine is looking to develop a state-of-the-art flight price prediction system which is able to predict the flight price/fare based on various factors such as travel date, destination and airlines. This system facilitates the users to access the portal anytime as long there is internet connectivity and predict the flight prices which in turn helps to plan their journey. At the same time, the airlines need not worry on investing on complex algorithms for their booking website.

Currently, airline companies use complex algorithms to calculate flight prices given various conditions present at that particular time. These methods take financial, marketing and various social factors into account to predict flight price. Also, the number of people using flights has increased significantly. It is difficult for airlines to maintain prices since prices change dynamically due to different conditions. For example, the flight prices with a layover have cheaper rates compared to flights with no layover. Flights scheduled in the morning are costlier compared to flights scheduled at night.

Due to these disadvantages of the current system, a flight price prediction system is proposed. Flight Price Prediction System is a machine learning based application which is based on various mathematical calculations. The system can be used to predict the flight prices on real time basis. The major advantages of this system are that you can access the portal anywhere in the world since the portal is hosted on a cloud platform which Google App Engine.

The model is trained on various data collected from various airline companies and a prediction model is built after analyzing the data. A front end is created for the user to interact that helps the user to predict the flight price. The flight fare price gets predicted after taking inputs from the user such as travel date, arrival date, airlines etc. Once the flight price is predicted the user can plan his trip. Since this website is hosted on a cloud system it can be accessed anywhere in the world with an internet connectivity. The cloud platforms such as Google App Engine, AWS provide an infrastructure to user to host such websites.

CHAPTER 2

LITERATURE REVIEW

2.1 PYTHON

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built-in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance.

Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed. Python is dynamically typed and garbage-collected. It supports multiple programming paradigms, including procedural, object-oriented, and functional programming. Python is a general purpose and high-level programming language. We can use Python for developing desktop GUI applications, websites and web applications. Also, Python, as a high-level programming language, allows you to focus on core functionality of the application by taking care of common programming tasks.

Modules in Python are simply Python files with a .py extension. The name of the module will be the name of the file. A Python module can have a set of functions, classes or variables defined and implemented. A module can contain executable statements as well as function definitions. These statements are intended to initialize the module. They are executed only the first time the module name is encountered in an import statement.

Features of Python are as follows:

- Easy to Learn and Use
- Expressive Language
- Interpreted Language
- Free and Open Source
- Object-Oriented Language
- Extensible
- Large Standard Library

Advantages of Python are:

- 1. Presence of Third-Party Modules:** The Python Package Index (PyPI) contains numerous third-party modules that make Python capable of interacting with most of the other languages and platforms.

- 2. Extensive Support Libraries:** Python provides a large standard library which includes areas like internet protocols, string operations, web services tools and operating system interfaces. Many high use programming tasks have already been scripted into the standard library which reduces length of code to be written significantly.

- 3. Open Source and Community Development:** Python language is developed under an OSI-approved open-source license, which makes it free to use and distribute, including for commercial purposes. Further, its development is driven by the community which collaborates for its code through hosting conferences and mailing lists, and provides for its numerous modules.

- 4. Learning Ease and Support Available:** Python offers excellent readability and uncluttered simple-to-learn syntax which helps beginners to utilize this programming language. The code style guidelines, PEP 8, provide a set of rules to facilitate the formatting of code. Additionally, the wide base of users and active developers has resulted in a rich internet resource bank to encourage development and the continued adoption of the language.

- 5. User-friendly Data Structures:** Python has built-in list and dictionary data structures which can be used to construct fast runtime data structures. Further, Python also provides the option of dynamic high-level data typing which reduces the length of support code that is needed.

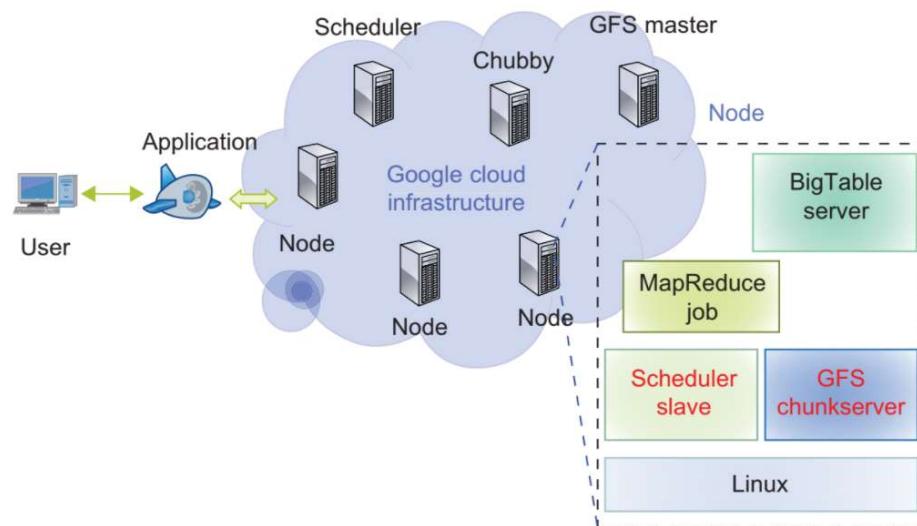
- 6. Productivity and Speed:** Python has cleaned object-oriented design, provides enhanced process control capabilities, and possesses strong integration and text processing capabilities and its own unit testing framework, all of which contribute to the increase in its speed and productivity. Python is considered a viable option for building complex multi-protocol network applications.

Disadvantages of Python are as follows:

1. Speed: Python is slower than C or C++.
2. Mobile Application Development: Python is not a very good language for mobile application development.
3. Memory Consumption: Python is not a good choice for memory intensive tasks.
4. Database Access: Python has limitations with database access. Runtime Errors are more.

2.2 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.



2.3 JUPYTER NOTEBOOK

Jupyter Notebook (formerly IPython Notebooks) is a web-based interactive computational environment for creating Jupyter notebook documents. The "notebook" term can colloquially make reference to many different entities, mainly the Jupyter web application, Jupyter Python web server, or Jupyter document format depending on context. A Jupyter Notebook document is a JSON document, following a versioned schema, containing an ordered list of input/output cells which can contain code, text (using Markdown), mathematics, plots and rich media, usually ending with the ".ipynb" extension.

A Jupyter Notebook can be converted to a number of open standard output formats (HTML, presentation slides, LaTeX, PDF, ReStructuredText, Markdown, Python) through "Download as" in the web interface, via the nbconvert library or "jupyter nbconvert" command line interface in a shell. To simplify visualization of Jupyter notebook documents on the web, the nbconvert library is provided as a service through NbViewer which can take a URL to any publicly available notebook document, convert it to HTML on the fly and display it to the user.

2.4 HTML

The Hypertext Markup Language, or HTML is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets and scripting languages such as JavaScript.

2.5 CSS

Cascading Style Sheets is a style sheet language used for describing the presentation of a document written in a markup language such as HTML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript.

2.6 FLASK

Flask is a micro web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions.

2.7 NUMPY

NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays. The ancestor of NumPy, Numeric, was originally created by Jim Hugunin with contributions from several other developers. In 2005, Travis Oliphant created NumPy by incorporating features of the competing Numarray into Numeric, with extensive modifications. NumPy is open-source software and has many contributors. NumPy is a NumFOCUS fiscally sponsored project.

2.8 PANDAS

Pandas is a software library written for the Python programming language for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series. It is free software released under the three-clause BSD license. The name is derived from the term "panel data", an econometrics term for data sets that include observations over multiple time periods for the same individuals. Its name is a play on the phrase "Python data analysis" itself. Wes McKinney started building what would become pandas at AQR Capital while he was a researcher there from 2007 to 2010.

2.9 SCIKIT LEARN

Scikit-learn is a free software machine learning library for the Python programming language. It features various classification, regression and clustering algorithms including support vector machines, random forests, gradient boosting, k-means and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy. Scikit-learn is a NumFOCUS fiscally sponsored project.

CHAPTER 3

REQUIREMENT SPECIFICATION

The hardware and software components of a computer system that are required to install and use software efficiently are specified in the SRS. The minimum system requirements need to be met for the programs to run at all times on the system.

3.1 HARDWARE REQUIREMENTS

The hardware requirements specify the necessary hardware which provides us the platform to implement our programs.

- 2.2 GHz processor (Pentium).
- GB RAM (System Memory).
- 20 GB of hard-drive space.
- VGA capable of 1024 x 768 screen resolution.
- Necessary computer peripherals such as keyboard etc.

3.2 SOFTWARE REQUIREMENTS

The software requirement specifies the pre-installed software needed to run the code being implemented in this project.

- Windows Operating System
- Google Cloud SDK
- Python 3.7

CHAPTER 4

DESIGN

A Flight Price Prediction System using Google App Engine is looking to develop a state-of-the-art flight price prediction system which is able to predict the flight price/fare based on various factors such as travel date, destination and airlines. This system facilitates the users to access the portal anytime as long there is internet connectivity and predict the flight prices which in turn helps to plan their journey. At the same time, the airlines need not worry on investing on complex algorithms for their booking website.

Currently, airline companies use complex algorithms to calculate flight prices given various conditions present at that particular time. These methods take financial, marketing and various social factors into account to predict flight price. Also, the number of people using flights has increased significantly. It is difficult for airlines to maintain prices since prices change dynamically due to different conditions. For example, the flight prices with a layover have cheaper rates compared to flights with no layover. Flights scheduled in the morning are costlier compared to flights scheduled at night.

Due to these disadvantages of the current system, a flight price prediction system is proposed. Flight Price Prediction System is a machine learning based application which is based on various mathematical calculations. The system can be used to predict the flight prices on real time basis. The major advantages of this system are that you can access the portal anywhere in the world since the portal is hosted on a cloud platform which Google App Engine.

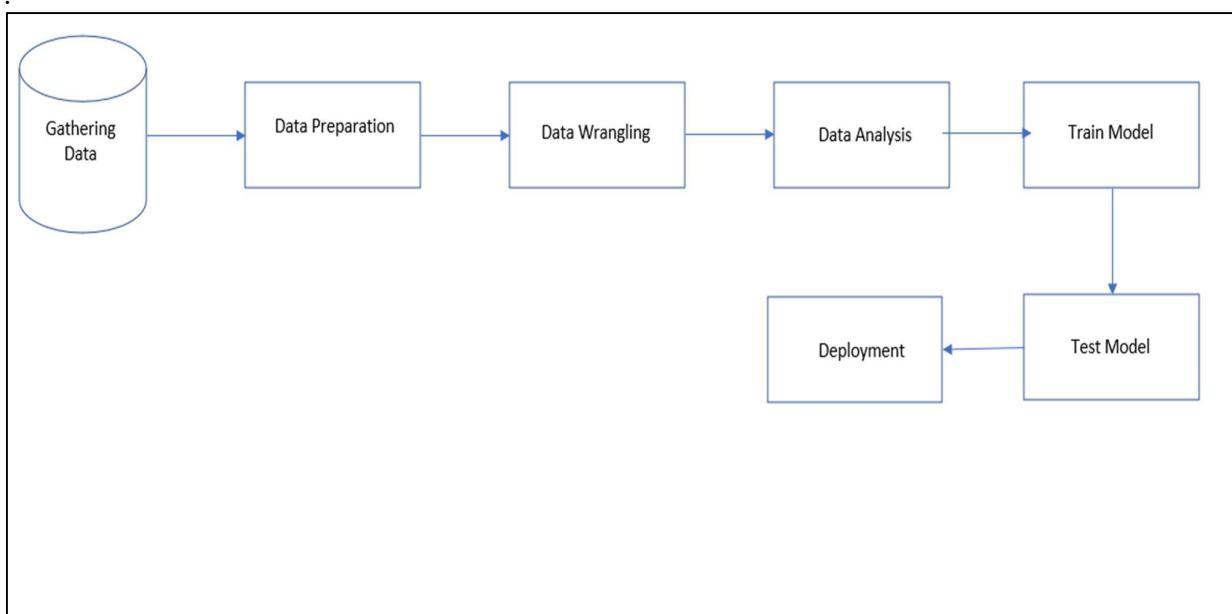


Fig 4.1 Design of the system – Flight Price Prediction

4.1 Data Gathering and Preparation:

Data collection is defined as the procedure of collecting, measuring and analysing accurate insights for research using standard validated techniques. A researcher can evaluate their hypothesis on the basis of collected data. In most cases, data collection is the primary and most important step for research, irrespective of the field of research. The approach of data collection is different for different fields of study, depending on the required information. The most critical objective of data collection is ensuring that information-rich and reliable data is collected for statistical analysis so that data-driven decisions can be made for research.

In our application the data is collected by open-source communities such as Kaggle and is curated and prepared for analysis in a structured format which can be used for predictive analysis. The data set for our application is collected from Kaggle which is an open-source community where users can get datasets for statistical analysis.

The dataset source link: <https://www.kaggle.com/nikhilmittal/flight-fare-prediction-mh/>

The train dataset consists of 10684 records with different columns such as date of journey, source destination, departure time, arrival time with a total of 11 columns. The test set contains around 2672 records.

4.2 Data Wrangling:

Data wrangling—also called data cleaning, data remediation, or data munging—refers to a variety of processes designed to transform raw data into more readily used formats. The exact methods differ from project to project depending on the data you’re leveraging and the goal you’re trying to achieve.

Some examples of data wrangling include:

- Merging multiple data sources into a single dataset for analysis
- Identifying gaps in data (for example, empty cells in a spreadsheet) and either filling or deleting them
- Deleting data that’s either unnecessary or irrelevant to the project you’re working on
- Identifying extreme outliers in data and either explaining the discrepancies or removing them so that analysis can take place

Data wrangling can be a manual or automated process. In scenarios where datasets are exceptionally large, automated data cleaning becomes a necessity. In organizations that employ a full data team, a data scientist or other team member is typically responsible for data wrangling. In smaller organizations, non-data professionals are often responsible for cleaning their data before leveraging it.

In our application the date column had to be converted from month-date-year format to date-month-format. Many categorical columns had to be converted to numerical columns for training a model as the model understands only numbers. The multiple columns recurring had to be removed.

4.3 Data Analysis:

Data analysis is defined as a process of cleaning, transforming, and modelling data to discover useful information for business decision-making. The purpose of Data Analysis is to extract useful information from data and taking the decision based upon the data analysis. A simple example of Data analysis is whenever we take any decision in our day-to-day life is by thinking about what happened last time or what will happen by choosing that particular decision. This is nothing but analysing our past or future and making decisions based on it. For that, we gather memories of our past or dreams of our future. So that is nothing but data analysis. Now same thing analyst does for business purposes, is called Data Analysis.

Here we plotted various graphs such as bar plots, correlation matrix etc to understand the relation between various columns. From the analysis we found that the flights with layover had a higher flight price. The flight prices of Jet Airways business class have a higher price. The flight price on Mondays is cheaper and the flights on Thursday are higher compared to all other days. The data analysis helps us understand the data in much better way.

4.4 Training the Model

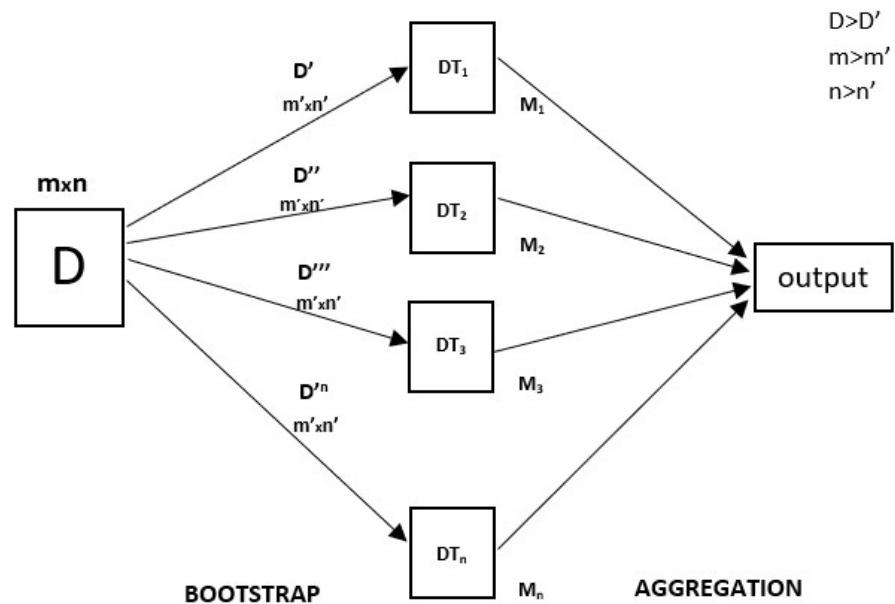
Model training is the phase in the data science development lifecycle where practitioners try to fit the best combination of weights and bias to a machine learning algorithm to minimize a loss function over the prediction range. The purpose of model training is to build the best mathematical representation of the relationship between data features and a target label (in supervised learning) or among the features themselves (unsupervised learning). Loss functions are a critical aspect of model training since they define how to optimize the machine learning algorithms. Depending on the objective, type of data and algorithm, data science practitioner use different type of loss functions. One of the popular examples of loss functions is Mean Square Error (MSE). In our system we have used random forest regressor algorithm for predicting the flight price prediction. The data is divided into training and test split for training the data and testing the data.

A Random Forest is an ensemble technique capable of performing both regression and classification tasks with the use of multiple decision trees and a technique called Bootstrap and Aggregation, commonly known as bagging. The basic idea behind this is to combine multiple decision trees in determining the final output rather than relying on individual decision trees.

Random Forest has multiple decision trees as base learning models. We randomly perform row sampling and feature sampling from the dataset forming sample datasets for every model. This part is called Bootstrap.

We need to approach the Random Forest regression technique like any other machine learning technique. The steps include:

- Design a specific question or data and get the source to determine the required data.
- Make sure the data is in an accessible format else convert it to the required format.
- Specify all noticeable anomalies and missing data points that may be required to achieve the required data.
- Create a machine learning model
- Set the baseline model that you want to achieve
- Train the data machine learning model.
- Provide an insight into the model with test data
- Now compare the performance metrics of both the test data and the predicted data from the model.
- If it doesn't satisfy your expectations, you can try improving your model accordingly or dating your data or use another data modeling technique.
- At this stage you interpret the data you have gained and report accordingly.



4.5 Test Model

Model Evaluation is an integral part of the model development process. It helps to find the best model that represents our data and how well the chosen model will work in the future. Evaluating model performance with the data used for training is not acceptable in data science because it can easily generate over optimistic and overfitted models. There are two methods of evaluating models in data science, Hold-Out and Cross-Validation. To avoid overfitting, both methods use a test set (not seen by the model) to evaluate model performance. In this method, the mostly large dataset is randomly divided to three subsets:

- 1.Training set is a subset of the dataset used to build predictive models.
- 2.Validation set is a subset of the dataset used to assess the performance of model built in the training phase. It provides a test platform for fine tuning model's parameters and selecting the best-performing model. Not all modelling algorithms need a validation set.
- 3.Test set or unseen examples is a subset of the dataset to assess the likely future performance of a model. If a model fit to the training set much better than it fits the test set, overfitting is probably the cause.

Cross-Validation

When only a limited amount of data is available, to achieve an unbiased estimate of the model performance we use k-fold cross-validation. In k-fold cross-validation, we divide the data into k subsets of equal size. We build models k times, each time leaving out one of the subsets from training and use it as the test set. If k equals the sample size, this is called "leave-one-out".

In our system used Mean Absolute Error, Mean Squared Error, Root Mean Squared Error as an evaluation metric to evaluate our system. The evaluation metric in our application is:

MAE: 1166.2496932991128

MSE: 4053539.8226505504

RMSE: 2013.3404636699056

4.6 Deployment

Software deployment includes all of the steps, processes, and activities that are required to make a software system or update available to its intended users. Today, most IT organizations and software developers deploy software updates, patches and new applications with a combination of manual and automated processes. Some of the most common activities of software deployment include software release, installation, testing, deployment, and performance monitoring.

Cloud deployment is the process of deploying an application through one or more hosting models—software as a service (SaaS), platform as a service (PaaS) and/or infrastructure as a service (IaaS)—that leverage the cloud. This includes architecting, planning, implementing and operating workloads on cloud.

CHAPTER 5

DEVELOPMENT

Flight Price Prediction.py

```
#!/usr/bin/env python  
# coding: utf-8
```

```
# In[216]:
```

```
import numpy as np  
import pandas as pd  
import matplotlib.pyplot as plt  
import seaborn as sns  
get_ipython().run_line_magic('matplotlib', 'inline')  
sns.set()
```

```
# In[217]:
```

```
train_data = pd.read_excel(r"Data_Train.xlsx")
```

```
# In[218]:
```

```
pd.set_option('display.max_columns', None)
```

```
# In[219]:
```

```
train_data.head()
```

```
# In[220]:
```

```
train_data.info()
```

```
# In[221]:
```

```
train_data["Duration"].value_counts()
```

```
# In[222]:
```

```
train_data.isnull().any()
```

In[223]:

```
train_data.isnull().sum()
```

In[224]:

```
train_data.dropna(inplace = True)
```

In[225]:

```
train_data.isnull().sum()
```

In[226]:

```
train_data["Journey_date"] = pd.to_datetime(train_data.Date_of_Journey,  
format = "%d/%m/%Y").dt.day  
train_data["Journey_month"] = pd.to_datetime(train_data["Date_of_Journey"], format =  
"%d/%m/%Y").dt.month  
train_data["Journey_day"] = pd.to_datetime(train_data["Date_of_Journey"], format =  
"%d/%m/%Y").dt.dayofweek
```

In[227]:

```
train_data.head()
```

In[228]:

```
train_data.drop(["Date_of_Journey"], axis = 1, inplace = True)
```

In[229]:

```
train_data["Dep_hour"] = pd.to_datetime(train_data["Dep_Time"]).dt.hour  
train_data["Dep_min"] = pd.to_datetime(train_data["Dep_Time"]).dt.minute
```

In[230]:

```
train_data.drop('Dep_Time', axis=1, inplace=True)
```

```
# In[231]:
```

```
train_data.head()
```

```
# In[232]:
```

```
train_data["Arrival_hour"] = pd.to_datetime(train_data.Arrival_Time).dt.hour  
train_data["Arrival_min"] = pd.to_datetime(train_data.Arrival_Time).dt.minute  
train_data.drop(["Arrival_Time"], axis=1, inplace=True)
```

```
# In[233]:
```

```
train_data.head()
```

```
# In[234]:
```

```
train_data.replace({"non-stop": 0, "1 stop": 1, "2 stops": 2, "3 stops": 3, "4 stops": 4}, inplace=True)
```

```
# In[235]:
```

```
train_data.head()
```

```
# In[236]:
```

```
duration = list(train_data["Duration"])
```

```
for i in range(len(duration)):
```

```
    if len(duration[i].split()) != 2: # Check if duration contains only hour or mins
```

```
        if "h" in duration[i]:
```

```
            duration[i] = duration[i].strip() + " 0m" # Adds 0 minute
```

```
        else:
```

```
            duration[i] = "0h " + duration[i] # Adds 0 hour
```

```
duration_hours = []
```

```
duration_mins = []
```

```
for i in range(len(duration)):
```

```
    duration_hours.append(int(duration[i].split(sep="h")[0])) # Extract hours from duration
```

```
    duration_mins.append(int(duration[i].split(sep="m")[0].split()[-1]))
```

```
# In[237]:
```

```
train_data["Duration_hours"] = duration_hours
```

```
train_data["Duration_mins"] = duration_mins
```

```
# In[238]:
```

```
train_data.drop(["Duration"], axis = 1, inplace = True)
```

```
# In[239]:
```

```
train_data.head()
```

```
# In[240]:
```

```
train_data["Airline"].value_counts()
```

```
# In[241]:
```

```
sns.catplot(y = "Price", x = "Airline", data = train_data.sort_values("Price", ascending = False),  
kind="boxen", height = 6, aspect = 3)  
plt.show()
```

```
# In[242]:
```

```
Airline = train_data[["Airline"]]  
Airline = pd.get_dummies(Airline,drop_first=True)  
Airline.head()
```

```
# In[243]:
```

```
Airline = train_data[["Airline"]]  
Airline = pd.get_dummies(Airline,drop_first=True)  
Airline.head()
```

```
# In[244]:
```

```
train_data['Source'].value_counts()
```

```
# In[245]:
```

```
sns.catplot(y = "Price", x = "Source", data = train_data.sort_values("Price", ascending = False))
```

```
plt.show()

# In[246]:
Source = train_data[["Source"]]
Source = pd.get_dummies(Source, drop_first= True)
Source.head()

# In[247]:
train_data["Destination"].value_counts()

# In[248]:
Destination = train_data[["Destination"]]
Destination = pd.get_dummies(Destination, drop_first = True)
Destination.head()

# In[249]:
train_data["Route"]

# In[250]:
train_data["Total_Stops"].value_counts()

# In[251]:
train_data.drop(["Route", "Additional_Info"], axis = 1, inplace = True)

# In[252]:
def time(row):
    if row>=21 or row<=18:
        return 0
    elif row>=6 or row<=12:
        return 1
    else:
        return 2
train_data['Time']=data_train['Dep_hour'].apply(time)
```

```
# In[253]:
```

```
train_data.head()
```

```
# In[260]:
```

```
train_data.Time.value_counts()
```

```
# In[267]:
```

```
fig, ax = plt.subplots(figsize=(10,5))
sns.countplot(x ='Time',hue='Airline', data = train_data,ax=ax)
```

```
# In[268]:
```

```
data_train = pd.concat([train_data, Airline, Source, Destination], axis = 1)
```

```
# In[269]:
```

```
data_train.head()
```

```
# In[270]:
```

```
data_train.drop(["Airline", "Source", "Destination"], axis = 1, inplace = True)
```

```
# In[257]:
```

```
data_train.head()
```

```
# In[271]:
```

```
sns.lineplot(data=data_train, x="Duration_hours", y="Price")
```

```
# In[272]:
```

```
sns.lineplot(data=data_train, x="Total_Stops", y="Price")
```

```
# In[273]:
```

```
fig, ax = plt.subplots()  
sns.lineplot(data=data_train, x="Journey_month", y="Price",ax=ax)  
ax.set_xlim(0,6)
```

```
# In[274]:
```

```
fig, ax = plt.subplots()  
sns.lineplot(data=data_train, x="Journey_day", y="Price",ax=ax)  
ax.set_xlim(0,6)
```

```
# In[275]:
```

```
data_train['Journey_month'].value_counts()
```

```
# In[276]:
```

```
data_train.shape
```

```
# In[277]:
```

```
data_train.head()
```

```
# In[278]:
```

```
test_data = pd.read_excel("Test_set.xlsx")
```

```
# In[279]:
```

```
test_data.head()
```

```
# In[280]:
```

```
# Preprocessing
```

```
print("Test data Info")  
print("-"*75)  
print(test_data.info())
```

```

print()
print()

print("Null values :")
print("-"*75)
test_data.dropna(inplace = True)
print(test_data.isnull().sum())

# EDA

# Date_of_Journey
test_data["Journey_date"] = pd.to_datetime(test_data["Date_of_Journey"],format="%d/%m/%Y").dt.day
test_data["Journey_month"] = pd.to_datetime(test_data["Date_of_Journey"],format="%d/%m/%Y").dt.month
test_data["Journey_day"] = pd.to_datetime(test_data["Date_of_Journey"],format="%d/%m/%Y").dt.dayofweek
test_data.drop(["Date_of_Journey"], axis = 1, inplace = True)

# Dep_Time
test_data["Dep_hour"] = pd.to_datetime(test_data["Dep_Time"]).dt.hour
test_data["Dep_min"] = pd.to_datetime(test_data["Dep_Time"]).dt.minute
test_data.drop(["Dep_Time"], axis = 1, inplace = True)

# Arrival_Time
test_data["Arrival_hour"] = pd.to_datetime(test_data.Arrival_Time).dt.hour
test_data["Arrival_min"] = pd.to_datetime(test_data.Arrival_Time).dt.minute
test_data.drop(["Arrival_Time"], axis = 1, inplace = True)

def time(row):
    if row>=21 or row<=18:
        return 0
    elif row>=6 or row<=12:
        return 1
    else:
        return 2
test_data['Time']=test_data['Dep_hour'].apply(time)

# Duration
duration = list(test_data["Duration"])

for i in range(len(duration)):
    if len(duration[i].split()) != 2: # Check if duration contains only hour or mins
        if "h" in duration[i]:
            duration[i] = duration[i].strip() + " 0m" # Adds 0 minute
        else:
            duration[i] = "0h " + duration[i] # Adds 0 hour

duration_hours = []
duration_mins = []
for i in range(len(duration)):
    duration_hours.append(int(duration[i].split(sep = "h")[0])) # Extract hours from duration
    duration_mins.append(int(duration[i].split(sep = "m")[0].split()[-1])) # Extracts only minutes from duration

```

```
# Adding Duration column to test set
test_data["Duration_hours"] = duration_hours
test_data["Duration_mins"] = duration_mins
test_data.drop(["Duration"], axis = 1, inplace = True)

# Categorical data

print("Airline")
print("-"*75)
print(test_data["Airline"].value_counts())
Airline = pd.get_dummies(test_data["Airline"], drop_first= True)

print()

print("Source")
print("-"*75)
print(test_data["Source"].value_counts())
Source = pd.get_dummies(test_data["Source"], drop_first= True)

print()

print("Destination")
print("-"*75)
print(test_data["Destination"].value_counts())
Destination = pd.get_dummies(test_data["Destination"], drop_first = True)

# Additional_Info contains almost 80% no_info
# Route and Total_Stops are related to each other
test_data.drop(["Route", "Additional_Info"], axis = 1, inplace = True)

# Replacing Total_Stops
test_data.replace({"non-stop": 0, "1 stop": 1, "2 stops": 2, "3 stops": 3, "4 stops": 4}, inplace = True)

# Concatenate dataframe --> test_data + Airline + Source + Destination
data_test = pd.concat([test_data, Airline, Source, Destination], axis = 1)

data_test.drop(["Airline", "Source", "Destination"], axis = 1, inplace = True)

print()
print()

print("Shape of test data : ", data_test.shape)

# In[281]:
```



```
data_test.head()
```



```
# In[282]:
```

```
data_train.shape
```

```
# In[283]:
```

```
data_train.columns
```

```
# In[284]:
```

```
X = data_train.loc[:,['Total_Stops', 'Journey_date', 'Journey_month', 'Journey_day',
'Dep_hour', 'Dep_min', 'Arrival_hour', 'Arrival_min', 'Duration_hours',
'Duration_mins', 'Airline_Air India', 'Airline_GoAir', 'Airline_IndiGo',
'Airline_Jet Airways', 'Airline_Jet Airways Business',
'Airline_Multiple carriers',
'Airline_Multiple carriers Premium economy', 'Airline_SpiceJet',
'Airline_Trujet', 'Airline_Vistara', 'Airline_Vistara Premium economy',
'Source_Chennai', 'Source_Delhi', 'Source_Kolkata', 'Source_Mumbai',
'Destination_Cochin', 'Destination_Delhi', 'Destination_Hyderabad',
'Destination_Kolkata', 'Destination_New Delhi','Time']]
```

```
X.head()
```

```
# In[285]:
```

```
X.shape
```

```
# In[291]:
```

```
X.head()
```

```
# In[286]:
```

```
y = data_train.iloc[:, 1]
y.head()
```

```
# In[287]:
```

```
plt.figure(figsize = (18,18))
sns.heatmap(train_data.corr(), annot = True, cmap = "RdYlGn")
plt.show()
```

```
# In[295]:
```

```
from sklearn.ensemble import ExtraTreesRegressor
selection = ExtraTreesRegressor()
selection.fit(X, y)
```

In[298]:

```
X.info()
```

In[303]:

```
print(len(selection.feature_importances_))
```

In[307]:

```
plt.figure(figsize = (12,10))
feat_importances = pd.Series(selection.feature_importances_, index=X.columns)
feat_importances.nlargest(25).plot(kind='barh')
plt.show()
```

In[308]:

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state = 42)
```

In[309]:

```
from sklearn.ensemble import RandomForestRegressor
reg_rf = RandomForestRegressor()
reg_rf.fit(X_train, y_train)
```

In[310]:

```
y_pred = reg_rf.predict(X_test)
```

In[311]:

```
reg_rf.score(X_train, y_train)
```

In[312]:

```
reg_rf.score(X_test, y_test)
```

```
# In[313]:
```

```
sns.distplot(y_test-y_pred)  
plt.show()
```

```
# In[314]:
```

```
plt.scatter(y_test, y_pred, alpha = 0.5)  
plt.xlabel("y_test")  
plt.ylabel("y_pred")  
plt.show()
```

```
# In[315]:
```

```
from sklearn import metrics
```

```
# In[316]:
```

```
print('MAE:', metrics.mean_absolute_error(y_test, y_pred))  
print('MSE:', metrics.mean_squared_error(y_test, y_pred))  
print('RMSE:', np.sqrt(metrics.mean_squared_error(y_test, y_pred)))
```

```
# In[317]:
```

```
metrics.r2_score(y_test, y_pred)
```

```
# In[318]:
```

```
from sklearn.model_selection import RandomizedSearchCV  
n_estimators = [int(x) for x in np.linspace(start = 100, stop = 1200, num = 12)]  
max_features = ['auto', 'sqrt']  
max_depth = [int(x) for x in np.linspace(5, 30, num = 6)]  
min_samples_split = [2, 5, 10, 15, 100]  
min_samples_leaf = [1, 2, 5, 10]
```

```
# In[319]:
```

```
random_grid = {'n_estimators': n_estimators,
```

```
'max_features': max_features,
'max_depth': max_depth,
'min_samples_split': min_samples_split,
'min_samples_leaf': min_samples_leaf}
```

In[320]:

```
rf_random = RandomizedSearchCV(estimator = reg_rf, param_distributions =
random_grid,scoring='neg_mean_squared_error', n_iter = 10, cv = 5, verbose=2, random_state=42,
n_jobs = 1)
```

In[321]:

```
rf_random.fit(X_train,y_train)
```

In[322]:

```
rf_random.best_params_
```

In[323]:

```
prediction = rf_random.predict(X_test)
```

In[324]:

```
plt.figure(figsize = (8,8))
sns.distplot(y_test-prediction)
plt.show()
```

In[325]:

```
print('MAE:', metrics.mean_absolute_error(y_test, prediction))
print('MSE:', metrics.mean_squared_error(y_test, prediction))
print('RMSE:', np.sqrt(metrics.mean_squared_error(y_test, prediction)))
```

In[326]:

```
import pickle
file = open('flight_rf.pkl', 'wb')
pickle.dump(reg_rf, file)
```

In[327]:

```
model = open('flight_rf.pkl','rb')
forest = pickle.load(model)
```

In[328]:

```
y_prediction = forest.predict(X_test)
```

In[329]:

```
metrics.r2_score(y_test, y_prediction)
```

Flask.py

```

from flask import Flask, request, render_template
from flask_cors import cross_origin
import sklearn
import pickle
import pandas as pd

app = Flask(__name__)
model = pickle.load(open("flight_rf.pkl", "rb"))

@app.route("/")
@cross_origin()
def home():
    return render_template("home.html")

@app.route("/predict", methods = ["GET", "POST"])
@cross_origin()
def predict():
    if request.method == "POST":

        # Date_of_Journey
        date_dep = request.form["Dep_Time"]
        Journey_date = int(pd.to_datetime(date_dep, format = "%Y-%m-%dT%H:%M").day)
        Journey_month = int(pd.to_datetime(date_dep, format = "%Y-%m-%dT%H:%M").month)
        Journey_day = int(pd.to_datetime(date_dep, format = "%Y-%m-%dT%H:%M").dayofweek)
        # print("Journey Date : ", Journey_day, Journey_month)

        # Departure
        Dep_hour = int(pd.to_datetime(date_dep, format = "%Y-%m-%dT%H:%M").hour)
        Dep_min = int(pd.to_datetime(date_dep, format = "%Y-%m-%dT%H:%M").minute)
        # print("Departure : ", Dep_hour, Dep_min)

        def time(row):
            if row >= 21 or row <= 18:
                return 0
            elif row >= 6 or row <= 12:
                return 1
            else:
                return 2
        Time = time(Dep_hour)

        # Arrival
        date_arr = request.form["Arrival_Time"]
        Arrival_hour = int(pd.to_datetime(date_arr, format = "%Y-%m-%dT%H:%M").hour)
        Arrival_min = int(pd.to_datetime(date_arr, format = "%Y-%m-%dT%H:%M").minute)
        # print("Arrival : ", Arrival_hour, Arrival_min)

        # Duration
        dur_hour = abs(Arrival_hour - Dep_hour)

```

```
dur_min = abs(Arrival_min - Dep_min)
# print("Duration : ", dur_hour, dur_min)

# Total Stops
Total_stops = int(request.form["stops"])
# print(Total_stops)

# Airline
# AIR ASIA = 0 (not in column)
airline=request.form['airline']
if(airline=='Jet Airways'):
    Jet_Airways = 1
    IndiGo = 0
    Air_India = 0
    Multiple_carriers = 0
    SpiceJet = 0
    Vistara = 0
    GoAir = 0
    Multiple_carriers_Premium_economy = 0
    Jet_Airways_Business = 0
    Vistara_Premium_economy = 0
    Trujet = 0

elif (airline=='IndiGo'):
    Jet_Airways = 0
    IndiGo = 1
    Air_India = 0
    Multiple_carriers = 0
    SpiceJet = 0
    Vistara = 0
    GoAir = 0
    Multiple_carriers_Premium_economy = 0
    Jet_Airways_Business = 0
    Vistara_Premium_economy = 0
    Trujet = 0

elif (airline=='Air India'):
    Jet_Airways = 0
    IndiGo = 0
    Air_India = 1
    Multiple_carriers = 0
    SpiceJet = 0
    Vistara = 0
    GoAir = 0
    Multiple_carriers_Premium_economy = 0
    Jet_Airways_Business = 0
    Vistara_Premium_economy = 0
    Trujet = 0

elif (airline=='Multiple carriers'):
    Jet_Airways = 0
    IndiGo = 0
    Air_India = 0
    Multiple_carriers = 1
    SpiceJet = 0
```

```
Vistara = 0
GoAir = 0
Multiple_carriers_Premium_economy = 0
Jet_Airways_Business = 0
Vistara_Premium_economy = 0
Trujet = 0

elif (airline=='SpiceJet'):
    Jet_Airways = 0
    IndiGo = 0
    Air_India = 0
    Multiple_carriers = 0
    SpiceJet = 1
    Vistara = 0
    GoAir = 0
    Multiple_carriers_Premium_economy = 0
    Jet_Airways_Business = 0
    Vistara_Premium_economy = 0
    Trujet = 0

elif (airline=='Vistara'):
    Jet_Airways = 0
    IndiGo = 0
    Air_India = 0
    Multiple_carriers = 0
    SpiceJet = 0
    Vistara = 1
    GoAir = 0
    Multiple_carriers_Premium_economy = 0
    Jet_Airways_Business = 0
    Vistara_Premium_economy = 0
    Trujet = 0

elif (airline=='GoAir'):
    Jet_Airways = 0
    IndiGo = 0
    Air_India = 0
    Multiple_carriers = 0
    SpiceJet = 0
    Vistara = 0
    GoAir = 1
    Multiple_carriers_Premium_economy = 0
    Jet_Airways_Business = 0
    Vistara_Premium_economy = 0
    Trujet = 0

elif (airline=='Multiple carriers Premium economy'):
    Jet_Airways = 0
    IndiGo = 0
    Air_India = 0
    Multiple_carriers = 0
    SpiceJet = 0
    Vistara = 0
    GoAir = 0
    Multiple_carriers_Premium_economy = 1
```

```
Jet_Airways_Business = 0
Vistara_Premium_economy = 0
Trujet = 0

elif (airline=='Jet Airways Business'):
    Jet_Airways = 0
    IndiGo = 0
    Air_India = 0
    Multiple_carriers = 0
    SpiceJet = 0
    Vistara = 0
    GoAir = 0
    Multiple_carriers_Premium_economy = 0
    Jet_Airways_Business = 1
    Vistara_Premium_economy = 0
    Trujet = 0

elif (airline=='Vistara Premium economy'):
    Jet_Airways = 0
    IndiGo = 0
    Air_India = 0
    Multiple_carriers = 0
    SpiceJet = 0
    Vistara = 0
    GoAir = 0
    Multiple_carriers_Premium_economy = 0
    Jet_Airways_Business = 0
    Vistara_Premium_economy = 1
    Trujet = 0

elif (airline=='Trujet'):
    Jet_Airways = 0
    IndiGo = 0
    Air_India = 0
    Multiple_carriers = 0
    SpiceJet = 0
    Vistara = 0
    GoAir = 0
    Multiple_carriers_Premium_economy = 0
    Jet_Airways_Business = 0
    Vistara_Premium_economy = 0
    Trujet = 1

else:
    Jet_Airways = 0
    IndiGo = 0
    Air_India = 0
    Multiple_carriers = 0
    SpiceJet = 0
    Vistara = 0
    GoAir = 0
    Multiple_carriers_Premium_economy = 0
    Jet_Airways_Business = 0
    Vistara_Premium_economy = 0
    Trujet = 0
```

```
# print(Jet_Airways,
#     IndiGo,
#     Air_India,
#     Multiple_carriers,
#     SpiceJet,
#     Vistara,
#     GoAir,
#     Multiple_carriers_Premium_economy,
#     Jet_Airways_Business,
#     Vistara_Premium_economy,
#     Trujet)

# Source
# Bangalore = 0 (not in column)
Source = request.form["Source"]
if (Source == 'Delhi'):
    s_Delhi = 1
    s_Kolkata = 0
    s_Mumbai = 0
    s_Chennai = 0

elif (Source == 'Kolkata'):
    s_Delhi = 0
    s_Kolkata = 1
    s_Mumbai = 0
    s_Chennai = 0

elif (Source == 'Mumbai'):
    s_Delhi = 0
    s_Kolkata = 0
    s_Mumbai = 1
    s_Chennai = 0

elif (Source == 'Chennai'):
    s_Delhi = 0
    s_Kolkata = 0
    s_Mumbai = 0
    s_Chennai = 1

else:
    s_Delhi = 0
    s_Kolkata = 0
    s_Mumbai = 0
    s_Chennai = 0

# print(s_Delhi,
#       s_Kolkata,
#       s_Mumbai,
#       s_Chennai)

# Destination
# Bangalore = 0 (not in column)
Source = request.form["Destination"]
if (Source == 'Cochin'):
```

```
d_Cochin = 1
d_Delhi = 0
d_New_Delhi = 0
d_Hyderabad = 0
d_Kolkata = 0

elif (Source == 'Delhi'):
    d_Cochin = 0
    d_Delhi = 1
    d_New_Delhi = 0
    d_Hyderabad = 0
    d_Kolkata = 0

elif (Source == 'New_Delhi'):
    d_Cochin = 0
    d_Delhi = 0
    d_New_Delhi = 1
    d_Hyderabad = 0
    d_Kolkata = 0

elif (Source == 'Hyderabad'):
    d_Cochin = 0
    d_Delhi = 0
    d_New_Delhi = 0
    d_Hyderabad = 1
    d_Kolkata = 0

elif (Source == 'Kolkata'):
    d_Cochin = 0
    d_Delhi = 0
    d_New_Delhi = 0
    d_Hyderabad = 0
    d_Kolkata = 1

else:
    d_Cochin = 0
    d_Delhi = 0
    d_New_Delhi = 0
    d_Hyderabad = 0
    d_Kolkata = 0

# print(
#     d_Cochin,
#     d_Delhi,
#     d_New_Delhi,
#     d_Hyderabad,
#     d_Kolkata
# )

# ['Total_Stops', 'Journey_day', 'Journey_month','Journey_day','Dep_hour',
# 'Dep_min', 'Arrival_hour', 'Arrival_min', 'Duration_hours',
# 'Duration_mins', 'Airline_Air India', 'Airline_GoAir', 'Airline_IndiGo',
# 'Airline_Jet Airways', 'Airline_Jet Airways Business',
# 'Airline_Multiple carriers',
```

```

# 'Airline_Multiple carriers Premium economy', 'Airline_SpiceJet',
# 'Airline_Trujet', 'Airline_Vistara', 'Airline_Vistara Premium economy',
# 'Source_Chennai', 'Source_Delhi', 'Source_Kolkata', 'Source_Mumbai',
# 'Destination_Cochin', 'Destination_Delhi', 'Destination_Hyderabad',
# 'Destination_Kolkata', 'Destination_New Delhi','Time',]

prediction=model.predict([[

    Total_stops,
    Journey_day,
    Journey_month,
    Journey_day,
    Dep_hour,
    Dep_min,
    Arrival_hour,
    Arrival_min,
    dur_hour,
    dur_min,
    Air_India,
    GoAir,
    IndiGo,
    Jet_Airways,
    Jet_Airways_Business,
    Multiple_carriers,
    Multiple_carriers_Premium_economy,
    SpiceJet,
    Trujet,
    Vistara,
    Vistara_Premium_economy,
    s_Chennai,
    s_Delhi,
    s_Kolkata,
    s_Mumbai,
    d_Cochin,
    d_Delhi,
    d_Hyderabad,
    d_Kolkata,
    d_New_Delhi,
    Time
]])

output=round(prediction[0],2)

    return render_template('home.html',prediction_text="Your Flight price is Rs.
{}".format(output))

return render_template("home.html")

if __name__ == "__main__":
    app.run()

```

Docker File

```
FROM continuumio/anaconda3:4.4.0
COPY . /usr/app/
EXPOSE 8000
WORKDIR /usr/app/
RUN pip install -r requirements.txt
CMD python app.py
```

CHAPTER 6

RESULTS

Model Evaluation is an integral part of the model development process. It helps to find the best model that represents our data and how well the chosen model will work in the future. Evaluating model performance with the data used for training is not acceptable in data science because it can easily generate over optimistic and overfitted models. There are two methods of evaluating models in data science, Hold-Out and Cross-Validation. To avoid overfitting, both methods use a test set (not seen by the model) to evaluate model performance.

We tried the application and received an 95% accuracy on the training dataset and approx. 80% accuracy on the test data. The model looks really quite well trained and can be used for prediction of flight prices based on various parameters.

The other evaluation metrics such as Mean Squared Error, Root Mean Squared Error etc. were also calculated and we received a marginal error.

MAE: 1166.2496932991128

MSE: 4053539.8226505504

RMSE: 2013.3404636699056

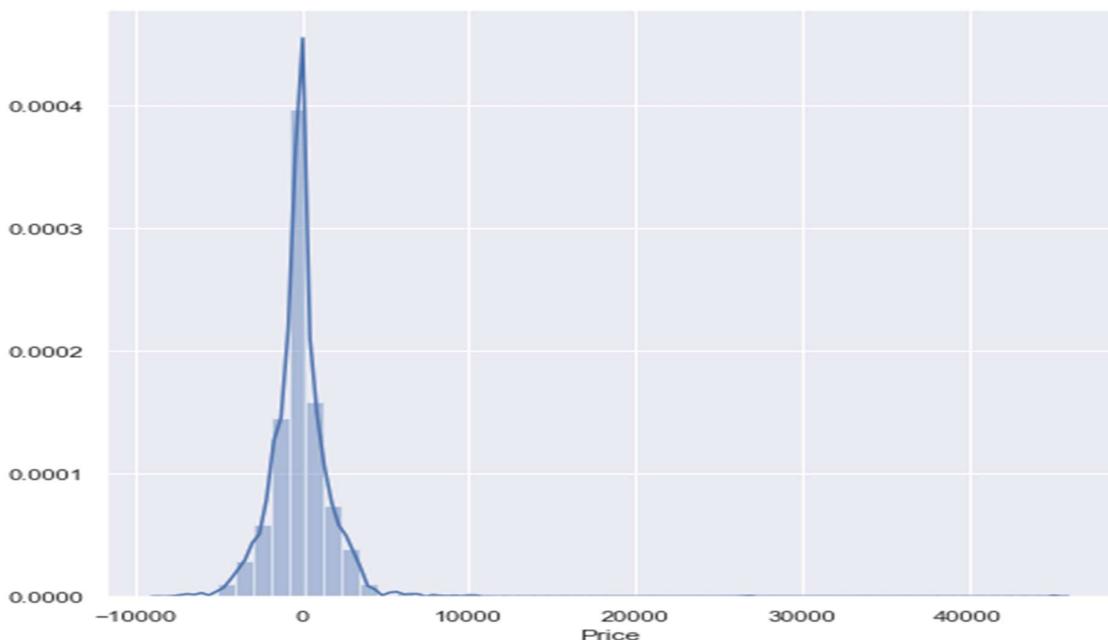


Fig 6.1 Denoting the Mean Absolute Error

The screenshot shows a web browser window titled "Flight Price Prediction" with the URL "127.0.0.1:5000/predict". The page has a light blue header bar with the title "FLIGHT PRICE". Below it is a form with the following fields:

- Departure Date:** A date input field with the placeholder "dd-mm-yyyy ::::".
- Arrival Date:** A date input field with the placeholder "dd-mm-yyyy ::::".
- Source:** A dropdown menu set to "Delhi".
- Destination:** A dropdown menu set to "Cochin".
- Stoppage:** A dropdown menu set to "Non-stop".
- Which Airline you want to travel?**: A dropdown menu with "Jet Airways" selected.
- Submit:** A button at the bottom of the form.

Below the form, a message says "Your Flight price is Rs. 11127.42". At the very bottom, there is a copyright notice: "©2020 Shaank M Kadwal".

Fig6.2 Front End

```

Command Prompt - python main.py
:My Files\CHPR\Projects\Flight Fare Prediction>python main.py
:Users\shash\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.9_qbzsn0kfr8pb\localCache\local_packages\Python39\site-packages\sklearn\base.py:310: UserWarning: Trying to unpickle estimator DecisionTreeRegressor from version 0.3.1 when using version 0.24.2. This might lead to breaking code or invalid results. Use at your own risk.
: warnings.warn(
:Users\shash\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.9_qbzsn0kfr8pb\localCache\local_packages\Python39\site-packages\sklearn\base.py:310: UserWarning: Trying to unpickle estimator RandomForestRegressor from version 0.3.1 when using version 0.24.2. This might lead to breaking code or invalid results. Use at your own risk.
: warnings.warn(
: Setting Fallback app "main" (lazy loading)
: Environment: production
: WARNING: This is a development server. Do not use it in a production deployment.
: Use a production WSGI server instead.
: Debug mode: off
: Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)

```

Fig 6.3 Deployment on Local Environment

CHAPTER 7

DEPLOYMENT

Software deployment includes all of the steps, processes, and activities that are required to make a software system or update available to its intended users. Today, most IT organizations and software developers deploy software updates, patches and new applications with a combination of manual and automated processes. Some of the most common activities of software deployment include software release, installation, testing, deployment, and performance monitoring.

Software development teams have innovated heavily over the past two decades, creating new paradigms and working methods for software delivery that are designed to meet the changing demands of consumers in an increasingly connected world. In particular, software developers have created workflows that enable faster and more frequent deployment of software updates to the production environment where they can be accessed by users.

7.1 CLOUD DEPLOYEMENT

Cloud deployment is the process of deploying an application through one or more hosting models—software as a service (SaaS), platform as a service (PaaS) and/or infrastructure as a service (IaaS)—that leverage the cloud. This includes architecting, planning, implementing and operating workloads on cloud.

Advantages of Cloud Deployment:

- Faster and simplified deployments. Automate builds that deploy code, databases and application releases, including resource provisioning.
- Cost savings. Control costs using consumption-based pricing and eliminate capex-heavy on-premises environments.
- Platform for growth. Leverage the global infrastructure provided by cloud service providers (CSPs) to seamlessly expand the business into other geographies.
- New digital business models. Exploit the continuous release of features and services by CSPs, incubate new technologies and innovate digital business models.
- Business resiliency. Architect for the availability and fault-tolerance CSPs offer and ensure disaster recovery and business continuity of applications to make the business resilient.
- Agility and scalability. Use autoscaling and scalability to meet peak demands of the business without provisioning for excess capacity.

- Geographic reach. Access applications from any location, on any device, leveraging the connectivity backbone of CSPs.
- Operational efficiency. Use the inherent automation enabled by cloud to increase operational efficiency and reduce human effort.
- A competitive edge. Leverage infrastructure as code and development, security and operations to reduce the time to market for new features and stay ahead of the competition.
- Empowered users. Increase productivity by empowering users with self-service options on cloud, such as portals, DevOps pipelines, and executive and operational dashboards.

For our application we will be using Google App Engine for deployment of the application which provides a platform as a service to host our applications.

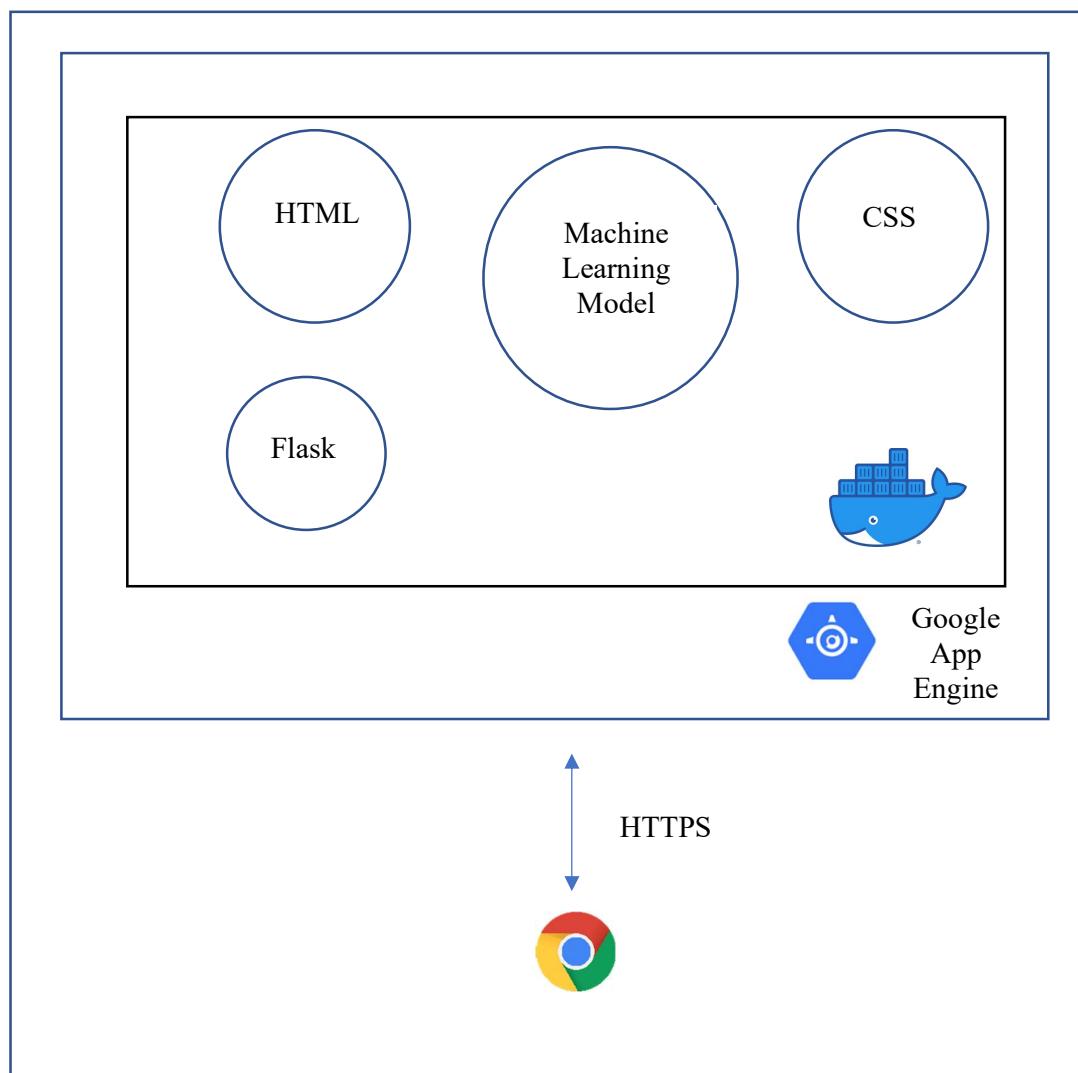


Fig 5.1 Deployment of the application on Google App Engine

7.2 Deploying on Google App Engine:

- Create a new Cloud Console project or retrieve the project ID of an existing project to use.
- Install and then initialize the Cloud SDK.
- Create an app.yaml file.
- Project Directory Structure:
 - app.yaml: Configure the settings of your App Engine application.
 - www/: Directory to store all of your static files, such as HTML, CSS, images, and JavaScript.
 - css/: Directory to store stylesheets.
 - style.css: Basic stylesheet that formats the look and feel of your site.
 - images/: Optional directory to store images.
 - index.html: An HTML file that displays content for your website.
 - js/: Optional directory to store JavaScript files.
 - Other asset directories.

Creating the app.yaml file

The app.yaml file is a configuration file that tells App Engine how to map URLs to your static files.

In the following steps, you will add handlers that will load www/index.html when someone visits your website, and all static files will be stored in and called from the www directory.

Create the app.yaml file in your application's root directory:

- Create a directory that has the same name as your project ID. You can find your project ID in the Console.
- In directory that you just created, create a file named app.yaml.
- Edit the app.yaml file and add the following code to the file:

```
runtime: python27
api_version: 1
threadsafe: true

handlers:
- url: /
  static_files: www/index.html
  upload: www/index.html

- url: /(.*)
  static_files: www/\1
  upload: www/(.*)
```

Deploying your application file:

When you deploy your application files, your website will be uploaded to App Engine. To deploy your app, run the following command from within the root directory of your application where the app.yaml file is located:

```
gcloud app deploy
```

Optional flags:

- Include the --project flag to specify an alternate Cloud Console project ID to what you initialized as the default in the gcloud tool. Example: --project [YOUR_PROJECT_ID]
- Include the -v flag to specify a version ID, otherwise one is generated for you. Example: -v [YOUR_VERSION_ID]

Viewing your application:

To launch your browser and view the app at https://PROJECT_ID.REGION_ID.r.appspot.com, run the following command:

```
gcloud app browse
```

7.3 Results and Dashboard

The screenshot shows a web browser window titled "Flight Price Prediction" with the URL "flightpriceprediction-336712.ey.r.appspot.com/predict". The page is titled "FLIGHT PRICE". It contains four input fields for "Departure Date" (14-01-2022 00:29), "Arrival Date" (14-01-2022 05:29), "Source" (Delhi), and "Destination" (Hyderabad). Below these are two more fields: "Stoppage" (Non-Stop) and "Which Airline you want to travel?" (Jet Airways). A "Submit" button is located below the "Stoppage" field. The result section displays the message "Your Flight price is Rs. 17402.43". At the bottom, there is a copyright notice: "©2020 Shashank M Kadiwal".

Fig 7.1 Front End Demonstration

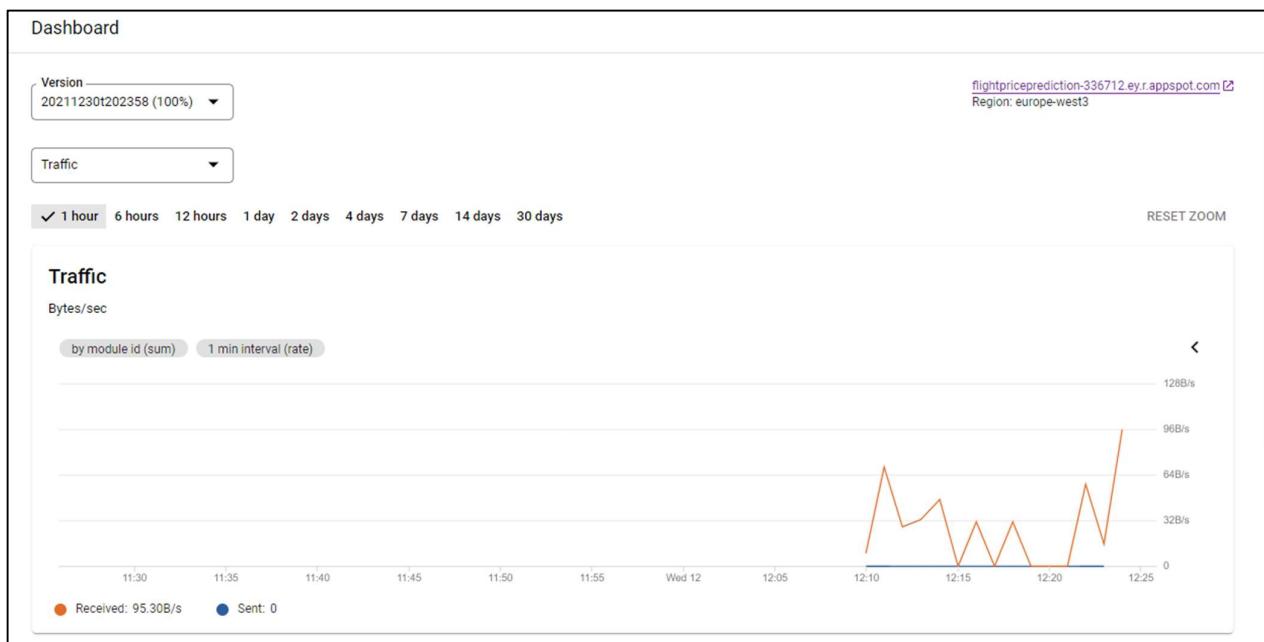


Fig 7.2 Dashboard Demonstrating the Traffic

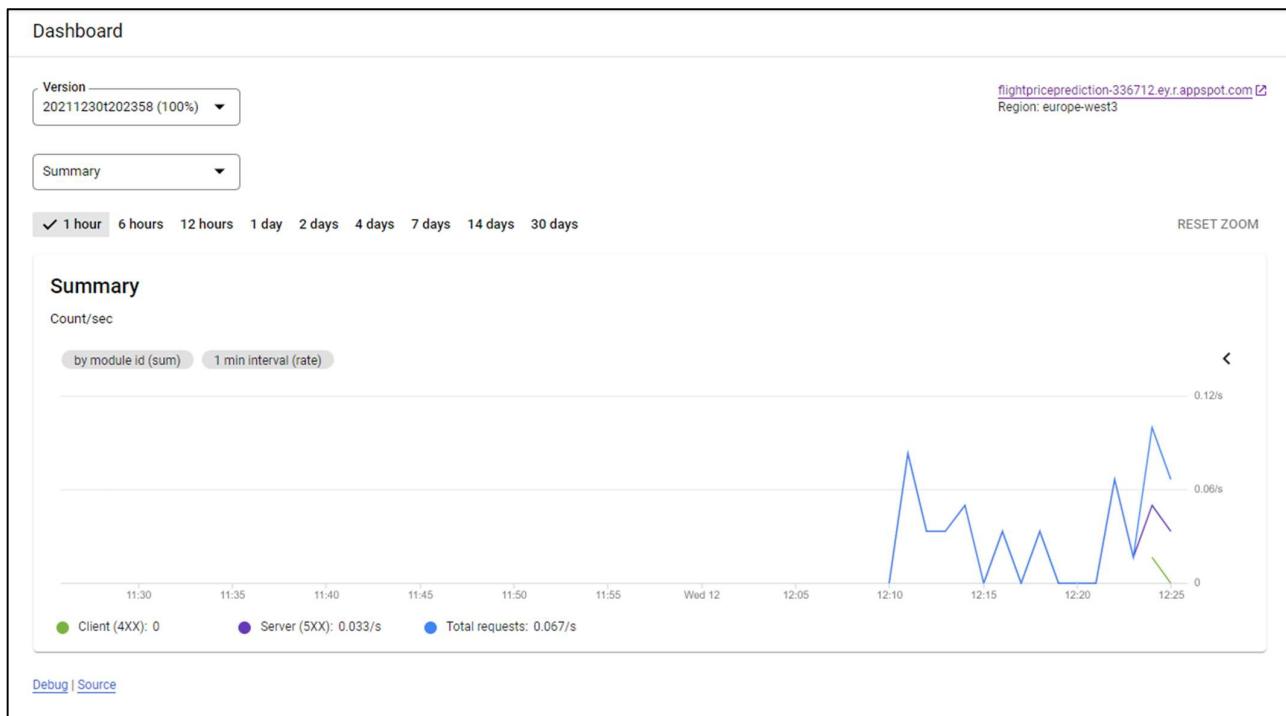


Fig 7.3 Dashboard Demonstrating the Summary



Fig 7.4 Dashboard Demonstrating the Error Details



Fig 7.5 Dashboard Demonstrating the Latency



Fig 7.5 Dashboard Demonstrating the Request Type

CONCLUSION

The project Flight Price Prediction is for computerizing the working in an airline company. The software takes care of all the requirements of an average airline company and is capable to predict the flight prices on various different conditions.

It takes into consideration to factors such as travel date, flight arrival date and also the existence of a layover. It provides the customer to predicted flight fares. The system can be accessed by anyone, anywhere since it is deployed on Google App Engine which provides PaaS.

BIBLIOGRAPHY

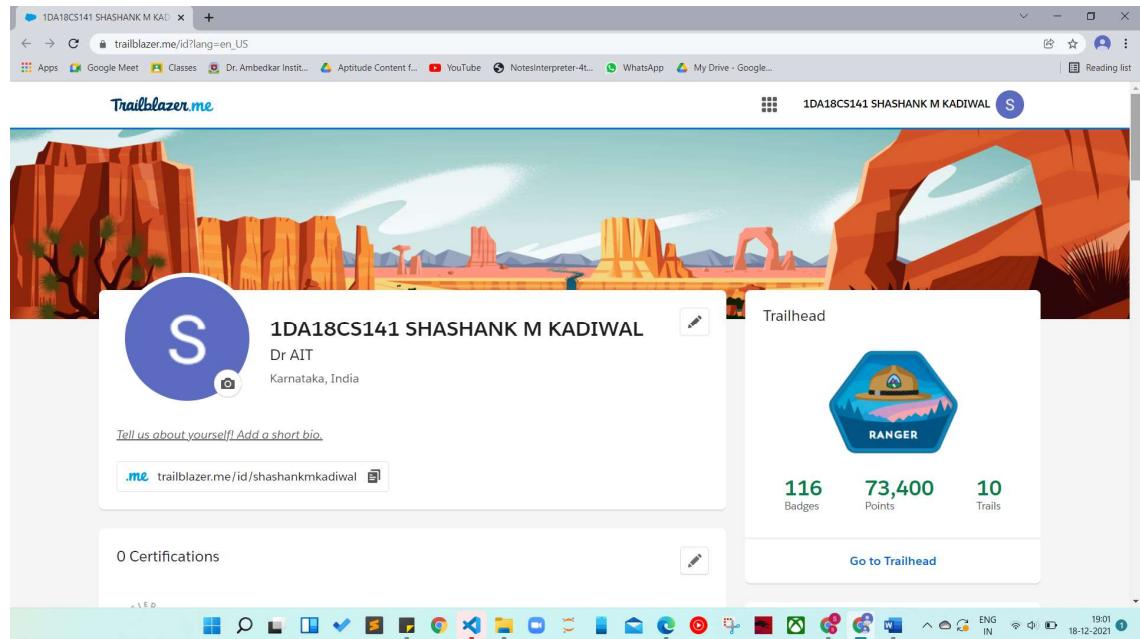
BOOK

1. Hands-On Machine Learning with Scikit-Learn and TensorFlow, Aurelien Geron
2. Distributed and Cloud Computing: From Parallel Processing to the Internet of Things, Kai Hwang
3. Introduction to Machine Learning with Python: A Guide for Data Scientists, Andreas C. Müller and Sarah Guido
4. Data Science on the Google Cloud Platform: Implementing End-to-End Real-Time Data Pipelines: From Ingest to Machine Learning, Valliappa Lakshmanan
5. Using Google App Engine, Charles Severance

URL

1. <https://www.youtube.com>
2. <https://www.google.co.in>
3. <https://www.wikipedia.org>
4. <https://cloud.google.com/appengine/docs/standard/python/getting-started/hosting-a-static-website>
5. <https://happyCoding.io/tutorials/java-server/hosting-google-app-engine>

INDUSTRY CERTIFICATIONS - SALESFORCE



A screenshot of a web browser showing a Salesforce Trailblazer profile. The profile belongs to user ID 1DA18CS141 SHASHANK M KADIWAL. The profile picture is a blue circle with a white letter 'S'. The user's name is listed as 1DA18CS141 SHASHANK M KADIWAL, Dr AIT, Karnataka, India. Below the profile picture is a text input field with placeholder text "Tell us about yourself! Add a short bio.". To the right of the profile is a "Trailhead" section featuring a hexagonal badge labeled "RANGER". Below the badge, statistics are displayed: 116 Badges, 73,400 Points, and 10 Trails. At the bottom right of the profile area is a "Go to Trailhead" button. The browser taskbar at the bottom shows various open tabs and system icons.