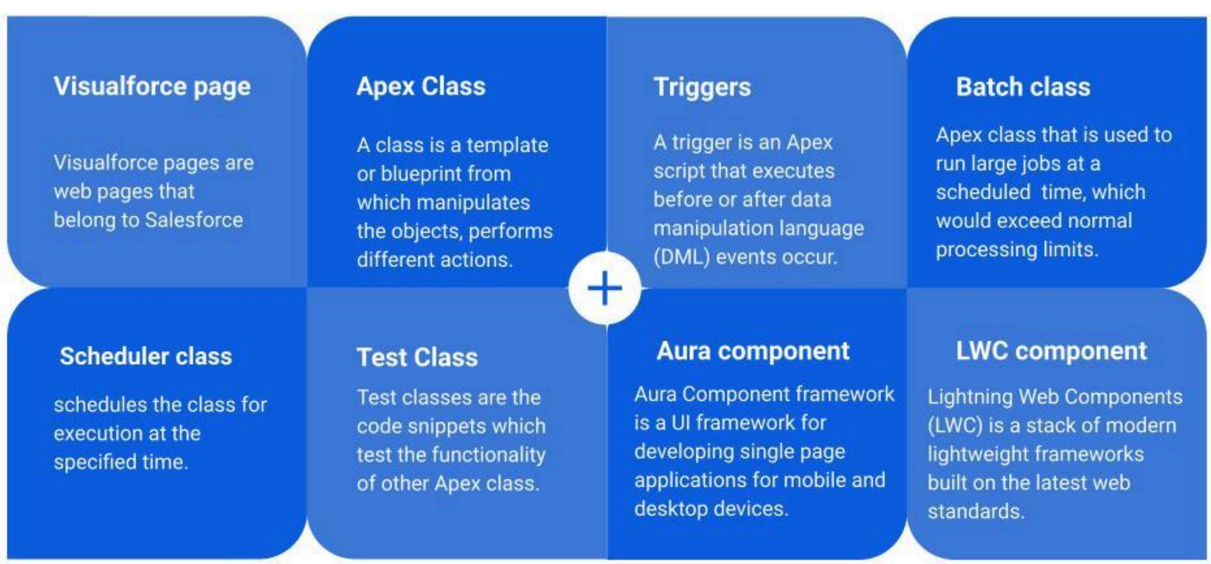
Salesforce Development Course

Pre Requirements

- Before you move forward with the development, I would suggest you do your development in VS Code. [Click here]
- **Note:** Check out how to set up VS Code for Salesforce and start your Development
- **Note:** Git & Github will be general (all the work should be saved on the GitHub repository)
- **Note:** Do Not use any AI tool (ex - ChatGPT, Bard) for development.



Salesforce Development

Development Course

1. Basics of Development
    1.1. Developer Console
    1.2. SOQL and SOSL
    1.3. Apex Basic
    1.4. Anonymous Window
    1.5. Debug Logs
2. Visualforce Pages
3. Apex Triggers and Apex Class
4. Exception Handling
5. Asynchronous Apex
    5.1. Batch Apex
    5.2. Future Methods
    5.3. Queueable Apex
    5.4. Scheduled Apex
6. Test Class
7. Workbench
8. Aura Lightning Component
9. Lightning Web Component
10. Integration
11. Test Class/Mock Class
12. Community
13. Deployment process along with the development
14. Salesforce Webservice
15. SOAP

1. Basics of Development

- Check Developer Console in org and read about that
- Read about SOQL and SOSL.
    - Get all contacts using SOQL in the query editor in the developer console
- Learn about apex
    - Learn about data types in Apex.
    - Learn about List, Set, and Map in Apex.
- Anonymous Window
    - Using Anonymous Windows get the list of all account
- Check debug logs after running the above code in Anonymous Window
- Setup debug logs from the setup.

2. Visualforce Pages

- Learn about Visualforce pages.
- Learn about different types of controllers in the Visualforce page.
- Create a Login page, Registration Page and Forgot Password Page
- Implement Pagination:
    - Display contact List with Pagination.
    - Create a vf page that user can select an account and it should display a list of opportunities related to that with Pagination.
- Display all Visualforce pages on the force.com site.

3. Apex Triggers

- Learn about Apex Triggers.
- How to write Apex Triggers?
- What is the use of apex triggers and how will they work?
- Learn about best practices for trigger and apex classes.

Task:

1. Write a trigger to display all trigger context variables using debug logs.
2. Write a trigger on Lead that will update rating as hot, whenever a new Lead record is created.
3. Write a trigger on opportunity that will set the stage name to “prospecting” and the close date to 15 days from today, Whenever the opportunity is modified.
4. Write a trigger on the opportunity that will set the opportunity type as a new customer.
5. Write a trigger so that whenever an account name is modified, send an email notification to the contact of an Account.
6. Write a trigger so that whenever an opportunity name is modified, create a task for the owner of the record.
7. Write a trigger to prefix the account name with Mr., or Mrs. whenever a new record is inserted.
8. Whenever contact is deleted, delete the corresponding account records.
9. Write a trigger so that whenever a new account is created, submit the record for approval.
10. Whenever a new account is created, create a new contact for the account with the same name as the contact.
11. Whenever a new contact is created, create an event related to that contact.
12. Write a trigger so that whenever a new record is created into an Account object before this new record is inserted delete the accounts with the same name.
13. When the rating of an account is modified to Hot, share the record with Wilson using apex-based sharing rules.

- **Complex triggers will be given below**

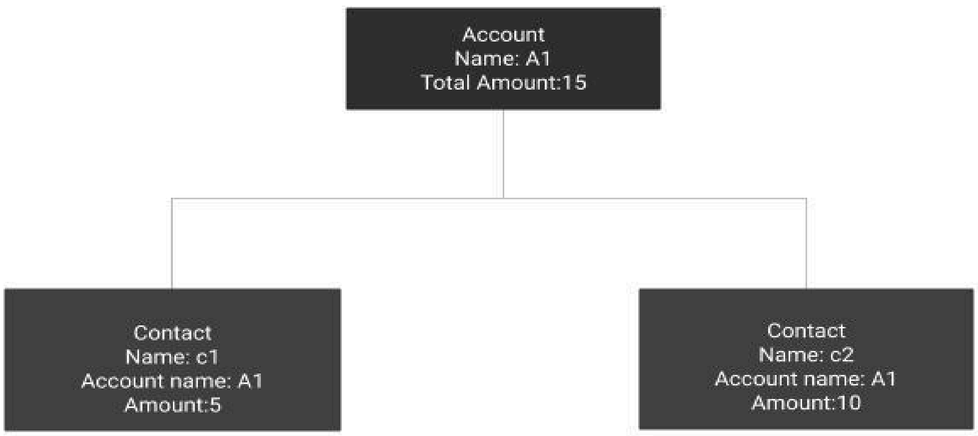
    1. Rollup summary
    2. sheep problem
    3. great problem

Rollup Summary

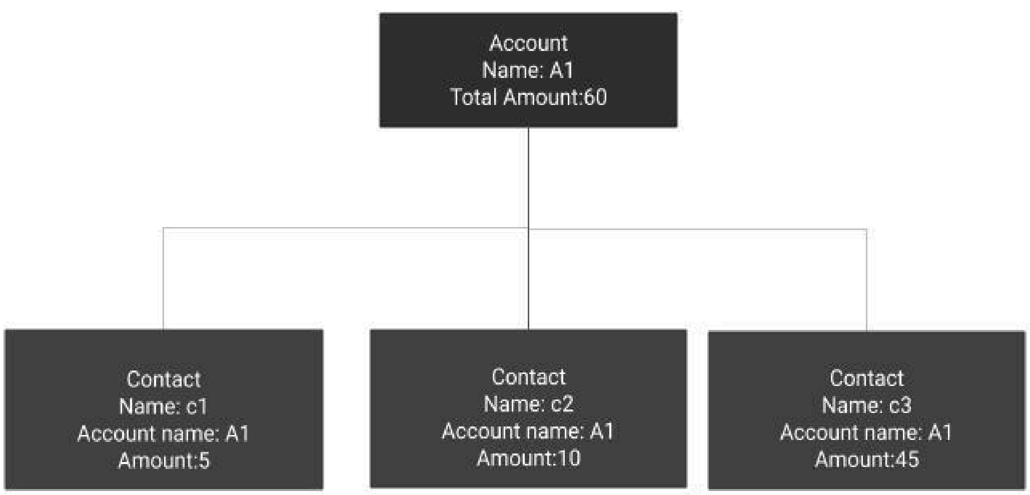
- Create a field - “Total Amount” in the Account Object, and another field - “Amount” in the Contact object. The “Total Amount” field will store the rollup of all the related contacts.

- Write a trigger so that whenever values are added/modified to the “Amount” field in Contact, It should update “Total Amount” to its associated parent Account.

- It should happen at the same time while deleting the contacts.



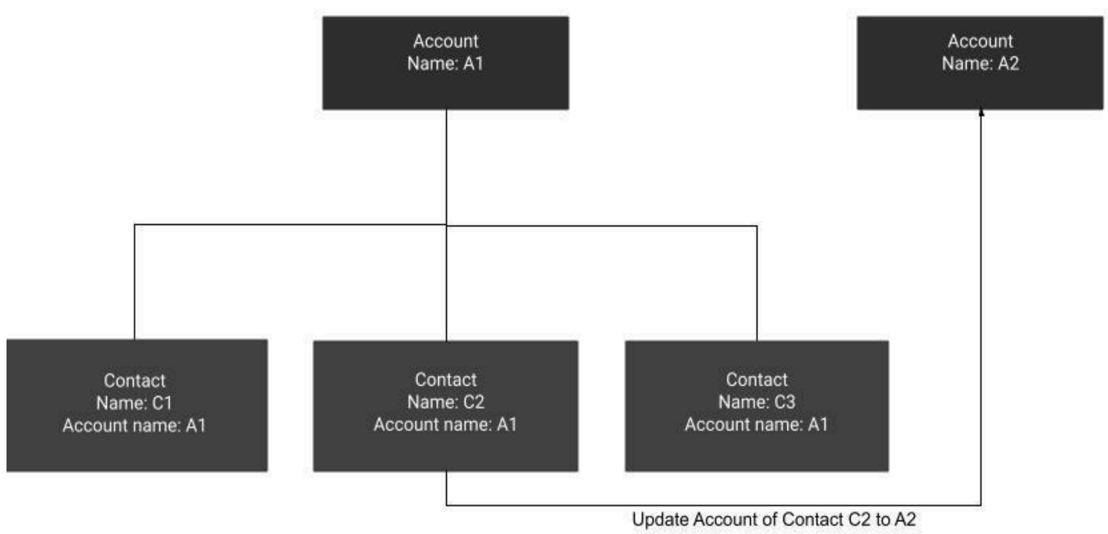
After Create/Update:



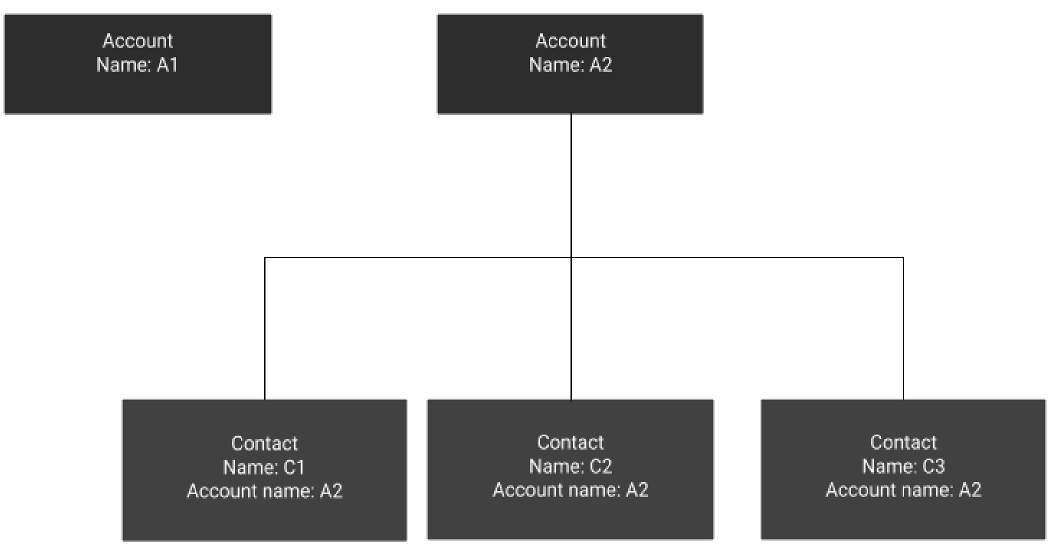
Sheep Problem

- Write a trigger so that whenever Contact is updated, It should update all contacts related to Its parent Account with the name of Account.

Before Update



After Update



The Great Problem

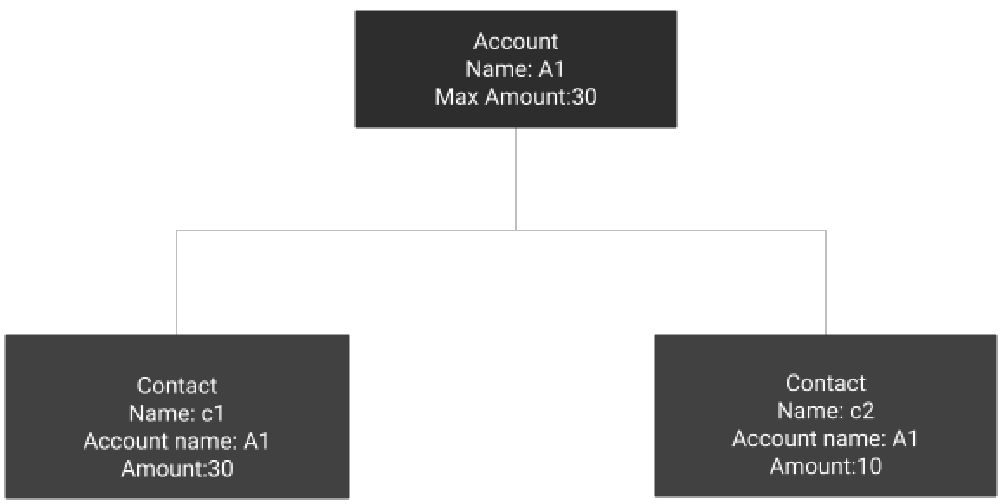
- Write a trigger so that whenever a Contact is created or updated, It should create/update the contacts of its related parent Account.

- Create a “Max Amount” field in the Account object, and another “Amount” field in Contact.

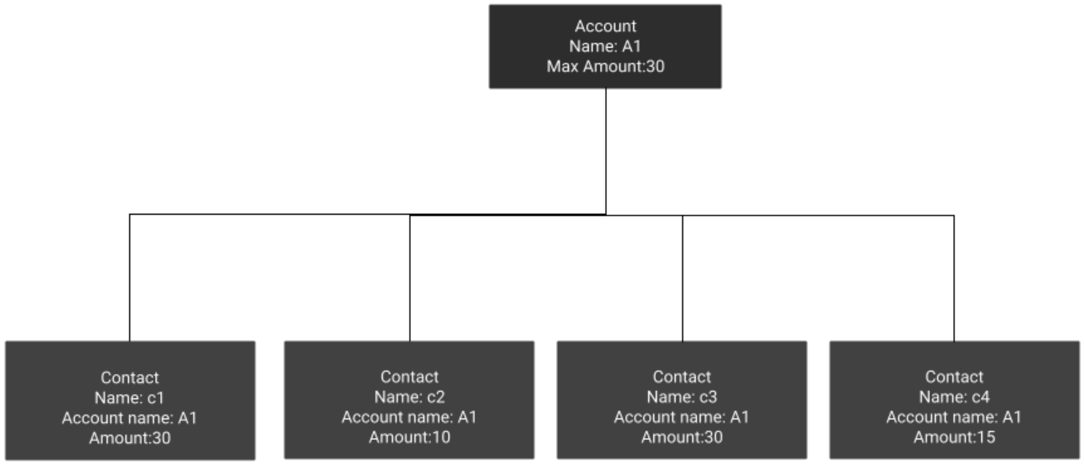
- So. Whenever a contact is created with the Amount field, It should check for the Max Amount of its parent account, If it’s lower than Amount It should simply insert the contact record. But, If It’s greater then It should divide that amount field value and contacts should get created with this divided amount.

- It should happen the same whenever a contact is created/Updated/Deleted.

An account with the Max_Amount: 30



After creating new Contact with the Amount of 45



It should create contact record with same amount as
Max amount and should create contact as per the remaining amount

4. Exception Handling

- Learn what is an exception.
- Learn about types of Exceptions.
- How to handle Exceptions in the apex class?

Task:

- Create custom object - **Error log** with the below fields
    - Class_Name__c (Text)
    - Exception_Message__c (Text Area)
    - Line_Number__c (Number)
    - Method_Name__c (Text)
    - Exception_Type__c (Text)
    - More_Details__c (Text Area)

- Create an Error log class which should contain the method that creates the Exception records.
- Implement the use of the Exception class in all rest of the tasks

5. Asynchronous Apex

5.1. Batch class

- What is the batch class?
- Why is the batch class needed?
- What are the limitations of batch class?

Task:

1. Create a batch class that will update the related account name with the opportunity name.
2. Create a batch class that will run every day at 9 AM and update all account description fields to the current date and time.

5.2. Future Apex

- What is Future Apex?
- Benefits of Future Apex.
- What are the Limitations of future apex?

Task:

1. Create a Button/Quick Action called Add Random File(Image) on the opportunity record page. So whenever I click on that button it should invoke an apex class which calls the future method and in that future method you have to use the web-service callout, the end result should have any random file(image) automatically attached to the attachment of that particular opportunity record.
   - E.g. open API - https://unsplash.com/documentation refer to this link for web-service callout or you can also use any other API as well.

5.3. Queueable Apex

- What is Queueable Apex?
- Benefits of Queueable Apex.
- Limitations of Queueable Apex.

5.4. Scheduled Apex

- What is Scheduled Apex?

- What is the use of schedule apex?
- Learn about Cron Expression.

Task:

1. Scheduled an apex class using a cron expression that will run every day at 11:30 AM and create 10 opportunity records.
2. Create a schedule that will run every week for any two days and update the rating field with “warm”.

6. Test Class

- Learn what is a test class.
- What is the purpose of writing a test class?
- What is the use of SeeAllData in the test class?
- Learn about different assert statements.

Task:

1. Write a test class for all the apex classes and triggers you have written. **(Note: Minimum code coverage should be 85%.)**

7. Workbench

- Learn about standard objects & Custom Objects in the workbench.
- Learn about Metadata types and components.
- Learn about Session Information.
- Learn what Password Management is.
- Learn what Bulk API job status shows.
- Learn what is Metadata API and process status.
- Learn what is the use of Apex execute in Workbench.

Task:

1. Execute a batch class using Apex execute.
2. Create 2000 records inside the Account object using Apex execute.

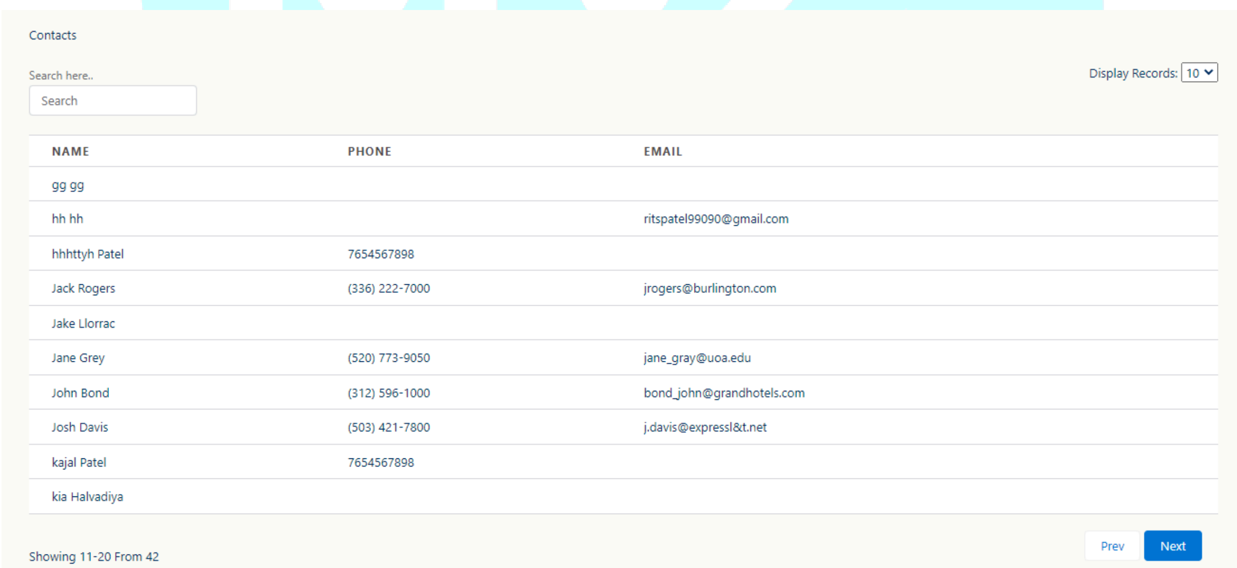
8. Aura Lightning Component

- What is the Lightning Aura component?
- Learn about the Aura component bundle.
- Why does the aura component need to be used over VisualForce pages?
- Learn about events.
- Learn about the use of ‘implements’ in the Lightning Aura component.
- Learn about Child to parent/Parent to child communication components.

Task:

1. Create an Aura component to create a contact record with an input field and take input from the user, ex: contact Name, mobile, etc.
    - Show toast message on successful record insert

2. Create the Aura component, which displays the contacts related to the account and should have pagination and search functionality.
    - Create an Aura Application for the same Aura component.



3. Drag & Drop functionality

i. Create an aura component, which should have two sections in the whole page, both of which should have the functionality of searching Accounts.

ii. Based on the search, It should display contacts related to the searched Account, and It should also display a count of contacts in both sections.

(Note: It's a must to get the number of contacts using Aggregate query).

iii. There should be functionality, like if someone drags the contact from one section to another, it should update the parent Account accordingly.

iv. Create a tab for the same Aura component in Salesforce.

4. Wizard using Lightning Aura component

i. Which should have next/previous buttons on each section, with the functionality of a progress bar.

ii. The first section should display a form for inserting an Account.

iii. On the next click, It should create an account and display a Contact Form.

iv. On clicking Next, It should create a contact related to the account created, and It should display a form to create an event.

v. There should be a button “Save” on the Create Event page, which should create an event for that contact with the Subject “Wizard Task”.

(Notes: The progress bar should be updated as per the visible form. on the First section Previous button should be disabled).

5. Change Path(Progress bar) using Custom Lightning Component on button click of next & previous.

(Notes: The progress bar should be updated as per the visible form. on the First section Previous button should be disabled).

6. Create a lightning component and add it to the home page. This component should be admin configurable where admin adds this component on the page admin should be able to see the following attributes to fill the value.

    - Image URL
    - Background Color
    - Description
    - Font size
    - Font Color

    - Once admin enters the value in this field then the component should render with a given value like image should be displayed as per the image URL, a description should be there under the Image with the mentioned font size and font color and also there should be background color as per the entered value. The concept is to make this component available on any page with our preferable value for the component.

    - Also, this component icon **should not** look like this 

    - Component icon should look like as per the below image:

7. Display list of all account records along with it display related opportunity and contact using the wrapper list and wrapper class.

8. Create an aura component, which contains a file uploader, whatever the image uploaded from the child component, should be previewed in the parent component.

9. Lightning Web Component

- What is the LWC Lifecycle?

- Lightning Message Services, PubSub model, LWC component bundles(Use of multiple js and css file in same LWC), Fire Event from LWC to Community Head markup
- File Upload Component for Lightning & Classic View Both with Uploaded Files table with pagination
- Learn about Lightning Message Service?

Task:

1. Create a LWC component to create a contact record with an input field and take input from the user, ex: contact Name, mobile, etc.
    - Show toast message on successful record insert

2. Create LWC component, which contains a file uploader, whatever the image uploaded from that component, should be previewed in the child component.

3. Create a LWC component, which should display a file uploader. Whenever someone uploads the file, All the uploaded files to that record should be display under the file uploader with Pagination.

    (Note: It should work for both Classic/Lightning view).

4. Create LWC component, which contains two different LWC components
    - Drop Down - which contains a list of Account, and on selection of Account Data should be passed to the other component using LMS(Lightning Message Service).
    - It should display a list of Contacts and Opportunities related to a Selected Account from the First LWC component.
    - Create a custom tab for the LWC component.

5. Create LWC wizard, Which should have 3 Sections with having Next/Previous button functionalities,
    - 1st Section - It should display a drop-down, containing 3 objects - Lead, Account, Contact. Based on selection It should display a list of records. It

should have the functionality of a checkbox at a record level. On click of Next, Next Section should be Visible.
    - 2nd Section - There should be an option to draft an email and on click of Next, the Next section should be Visible
    - 3rd Section - Email should be previewed and On click of Send, EMail should be sent to the selected Contact.

   (Note: Create custom email field, If not exists in any of the above Objects. Apply proper validation for the Next/Previous buttons).

6. Create LWC component which should display,
    - search bar
    - Multipicklist that should display a list of objects
    - Search button
    - On click of search button, It should search from the selected object records, and display all the data object wise

   (Note: It is must to use a SOSL to search in the object records)

10. Integration

- What is Integration? Why do we need to integrate two systems?
- What are the benefits of it?
- What is authentication and authorization in integration?
- What is named credentials and oAuth 2.0 in Salesforce?
- Learn about Connected App, How to authorize with Salesforce using connected app?

(Note: It is necessary to use Workbench for the API calling).

<u>Task:</u>

1. Gmail Integration - Integrate Gmail service with Salesforce
    - You have to create one button on contact record page called send email - so whenever I click on that button one popup component should be displayed in which user should be able to draft an email with attachment files and when user click on the send button, particular contact should get the email and this whole email sending process should be done via gmail integration.

2. Dropbox/AWS/OneDrive Integration:- you have to create one tab contact record page Called DropBox, it should be a custom component you can create either aura or LWC component as per your comfort it should let the user upload images or files or media files to the dropbox in which also they should be able to delete the Files as well

   (Note:- Here you can choose whatever service you like e.g. DropBox/AWS/OneDrive whatever you like).

3. Need to create a Salesforce API and test that API using postman, using that API we should be able to access any records of an object in Salesforce.
   - You can also use the CURL CLI tool for API testing.

11. Test Class/Mock Class

- What is Mock class? What is the use of it?

<u>Task:</u>

- Write a test class for all the apex classes you have written for Aura, LWC and Integration task..
- Write a Mock class for the REST/SOAP APIs you have created.

(Note: Minimum code coverage should be 85%).

12. Community

Create a Portal for Teachers and Students.

- Create a registration page that will allow you to select the first Teacher or Student. Based on this selection, fill in the appropriate fields in the Registration Form and Create a Community User. (Also create a new template for Community New User Welcome Email.)

- Create a Login page that will allow a teacher or Student to log into the Community Portal.

- Create a Forgot Password Page on which will send out emails based on the Username. (Also create a new template for Community Forgot password Email.)
- Create a Reset Password page that will allow Teacher/Student to reset their password. (Also create a new template for the community reset password Email.)
- Create Custom Page Not Found (Make it as unique as possible.)
- Create a new Custom Community Theme layout. Also in the community theme, add a menu section that will contain Home, My Profile, and Logout buttons that will navigate to the appropriate page.
- Change the Community Favicon Icon and Community title so that they are consistent for all of the pages.
- Create a Header and Footer Component that will be visible on all other pages (except the login and Registration pages).
- Create a home page using Standard Component Content (Like: HTML Editor, Rich Content Editor, Buttons, Horizontal Line, etc.).
- When logged in as Teacher,
  - Create one tab, “My Profile,” for the teacher where she can see the details. (Also, the Edit button, once clicked, will allow you to change the field data.)
  - Create another tab, “Students” for teachers where you can be able to see all the students listed under that login teacher. (Also once clicked on any Student it will display the Student detail page.)
  - Create a feedback button on the student detail page where teachers can give feedback using emojis and comments for individual Students. (This provided feedback will be stored in the Salesforce custom object.)
- When login as Student,
  - Create one tab “My Profile” for the Teacher where you can see the details. (Also give the Edit button once clicked it will allow you to change the field data.)
  - Create another tab “Class Semester” for Students where you can see the current Semester with all the subjects as a list view in it.
- Whenever Teacher and Student navigate to Home page then using community head markup fire an event that will display a popup box saying welcome message. (Create a custom and visually amazing Welcome message box.)
  (Session Storage)

- Add a Help page on both sides (Teacher/Student) that will be containing details regarding the community portal in it. It should be displayed using multiple languages. (Hint: Custom labels, Custom Metadata Types, Translation Workbench)

- Create a "Contact Us" Page in which the user can fill out a form with necessary details and it will send out an email to the Mentors email (MV Clouds Mentors). (Use an Aura component that will display screen flow which will provide all Form fields.)

- Use your own imagination to create any further pages/tabs in the portal.😇

13. Deployment process along with the development

- Learn what is Changeset.
- How deployment can be done using changeset?
- What are inbound and outbound changesets in Salesforce?
- Learn What is a workbench? How to deploy components using workbench?
- How to validate changeset and how to deploy it?

Task:

1. Create an account in Salesforce enterprise edition and then you have to create a sandbox.
2. In Sandbox add your any aura component task there (simply copy and paste your existing code there) and then test it in sandbox.
3. If it is working as expected then deploy this task in the production org using changeset. And test it in production. It should work fine in the production environment.

14. Salesforce Webservice

- What is the Salesforce webservice?
- Why do we need to use the Salesforce webservice?
- Learn about cURL and POSTMAN

Task:

1. Create one apex class in your org which can be called as a REST Service to the external/third party app.

- Add multiple methods to it which can perform different actions for example
    - Read (GET)
    - Create (POST)
    - Delete (DELETE)
    - Upsert (PUT)
    - Update (PATCH)

  (Note: During the time when you send a request you can send whatever (JSON data) body data that you want you can set according to your need.

- You have to test your method by using Workbench for REST APIs.
- Also you have to test your method by using **POSTMAN** or **cURL**

2. Create another apex class with exposing Apex Methods as SOAP Web Services. In response to this method you should get the account's related child records.

15. SOAP

- What is SOAP? Learn about SOAP API.
- What is WSDL?
- Learn about the types of WSDL.
- What is the use of SOAP and WSDL?
- What is the difference between REST and SOAP API.

Task:

1. Configure your Salesforce org using wsdl file to use SOAP api. Also do all the additional steps required to use SOAP api.
2. Create 10 account records with name 'TestAccount' + indexnum(1,2..10) using soap api.
3. Create 10 account records with name 'TestContact' + indexnum(1,2..10) using soap api.
4. Delete the contact and account records using SOAP api.