Requirement - 1

* Create module named res\_partner\_demo\_ksc
* Create a model which holds the data of customers.
* Name of the model will be res.partner.demo1
* Fields of the model will be as follows:
  + name - Char
  + email - Char
  + street1 - Char
  + street2 - Char
  + city - Char
  + state - Char
  + zip\_code - Char
  + country - Char
  + birthdate - Date
  + age - Integer
  + weight - Float
  + description - Text
  + gender - Selection(Male, Female, Transgender)
  + details - HTML
  + is\_spectacles - Boolean
  + photo - Image
* Create Tree, Form, Search View
* Search view
  + one should be able to search by name, city, state, country, zipcode
  + one should be able to group by country, state, city, gender
  + add filters of Male, Female, Transgender
  + add filter of Having Spectacles
* Create Main menu as Contacts
* Create sub menu as Customers
* Create sub menu as Customers
* Create appropriate action
* Create appropriate security
* Use “User : Own Documents Only”
* Add demo data to your module - 5 records

Requirement - 2

* Create new module - res\_country\_ksc
* Model - res.country.ksc
* Fields -
  + Name of the country - Char
  + Short Code of the country - Char
  + Active - Boolean
* Define Action
* Menu Name - Countries
  + Parent Menu - Localization
* View - Tree, Form, Search
* Data - Enter few data using xml using record tag
* Search View
  + One should be able to search by country name
  + One should be able to search by country code
  + Add filter of Active and Inactive
* Security - Add appropriate security from sales\_team module

Requirement - 3

* Create new module - res\_state\_ksc
* Model - res.state.ksc
* Fields -
* Name of the state - Char
* Short Code of the state - Char
* Country Name - Char - should not be copied during duplicate record
* Active - Boolean
* Define Action
* Menu Name - States
  + Parent Menu - Localization
* View - Tree, Form, Search
* Data - Enter few data using xml using record tag
* Search View
  + One should be able to search by state name
  + One should be able to search by state code
  + Group by Country Name
  + Add filter of Active and Inactive
* Security - Add appropriate security from sales\_team module

○

Requirement - 4

* Create new module - res\_city\_ksc
* Model - res.city.ksc
* Fields -
  + City Name - Char
  + State Name - Char - should not be copied during duplicate record
  + Active - Boolean
* Define Action
* Menu Name - City
  + Parent Menu - Localization -> Localization
* View - Tree, Search
* Tree - Make tree editable, we don’t need form view
* Search View
  + One should be able to search by city name
  + One should be able to search by state
  + Add filter of Active and Inactive
* Security - Add appropriate security from sales\_team module

Requirement - 5

Create new module named product\_ksc

Model - product.ksc

Fields -

1. Name of the product - Char
2. SKU(Stock Keeping Unit) of the product - Char
3. Barcode of the product - Char
4. Can this product be sold - Boolean
5. Product Type - (Storable, Consumable, Service) - Radio Type, Selection
6. Sale Price - Float, digits(6,2)
7. Cost Price - Float, digits(6,2)
8. Active - Boolean
9. Warehouse - Char
10. Product Image - image
11. Website Description - HTML
12. Internal Note - Text

Define Action

Menu Name - Products

Parent Menu - My Sales -> Products

View - Tree, Form, Search

Tree View - Product Name, SKU, Barcode, Product Type

Search View - Product Name, SKU, Barcode, Can this product be sold, Product Type

Requirement - 6

Create new module employee\_ksc

Model - employee.ksc

Fields -

1. Name of the employee - Char
2. Department Name of the employee - Char
3. Job Position of employee - Char
4. Salary - Float, digits(6,2)
5. Hire Date - Date
6. Gender - (Male, Female, Transgender)- Selection
7. Job Type - (Permanent, Ad Hoc) - Selection Define Action

View - Tree, Form, Search Search -

Employee Name, Job Position, Department Name Filter - Job Type (Show example of domain)

Filter - Gender

Group By - Job Position, Department Name, Job Type, Gender Security - Create separate security groups for user and manager

Requirement - 7

* Create new module named crm.lead.ksc
* Read about the what is a lead in google
* Fields :
  + Customer name - Char required
  + Customer email - Char required
  + Customer phone - Char required
  + Expected revenue - Float with 3 decimals
  + Salesperson - Char required
  + Sales Team - Char
  + Campaign - Char
  + Channel - Selection (Facebook, WhatsApp, Email, Newspaper, Television, Phone Call, SMS, Google Ads, ) - required
  + State - Selection (New, Qualified, Proposition, Won, Lost) - use statusbar widget with clickable option
  + Next Follow Up Date - Date - required
  + Won Date - make it visible only when the state is ‘won’ and make it required too
  + Lost reason - make it visible only when the state is ‘lost’ and make it required too
* Views -
  + Tree, Form, Search
  + Search - Customer name, email, phone
    - Group by - Channel
    - On Search view add a filter of today which will list based on the ‘next follow up date’ which are today - search in core code of odoo and documentation for that
  + Tree - search in documentation and core code
    - The leads which are Lost should be shown in Red colour
    - The leads which are Won should be shown in Green colour
    - The leads which are Proposition should be in orange colour
* Security - Add new security group for user and manager
  + Add a record rule where users from “Salesperson” group should only be able to view records of channel “Newspaper, Television, Phone Call, SMS”, while “Salesmanager” should be able to see all the records