Assignment
Custom Entity

ques 1:- Q1. Create a Custom Entity having the following fields :

•Name

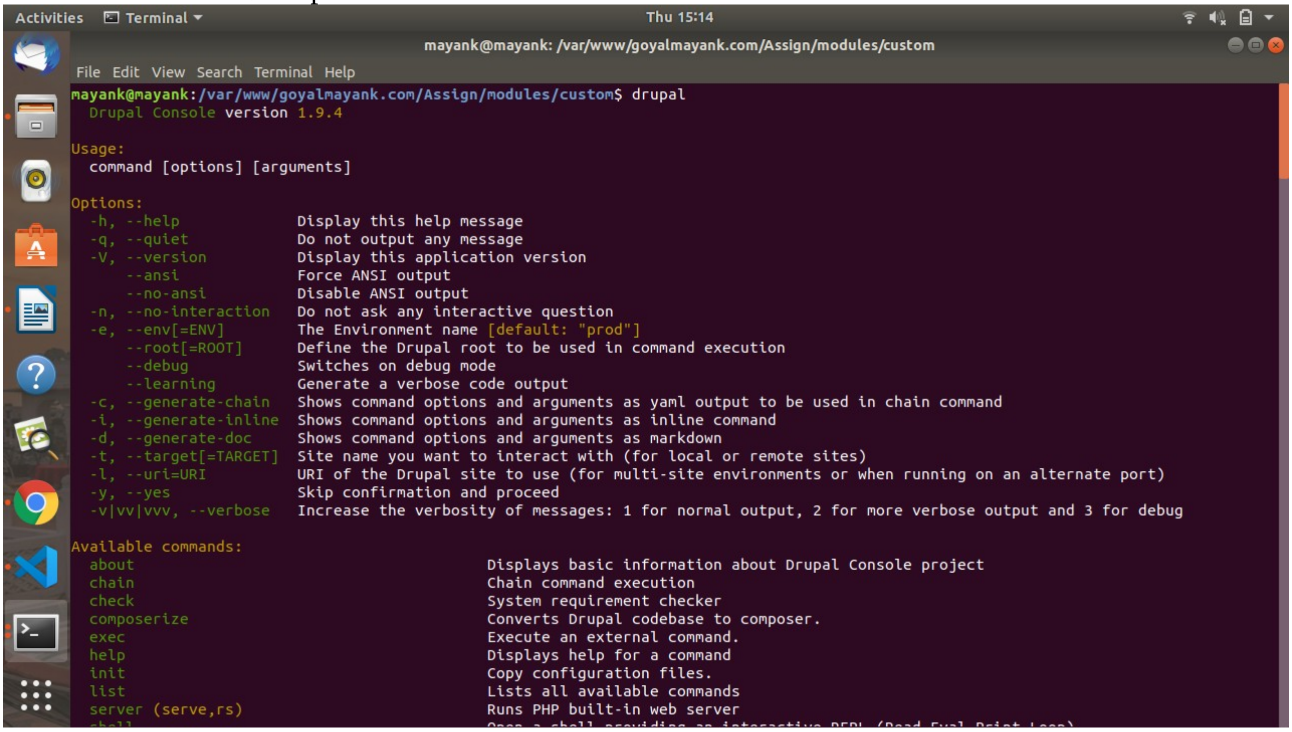
•Email

•Contact number

Soln :- step 1 : install a drupal console

step2 : - with help of terminal go to your drupal website and type the command

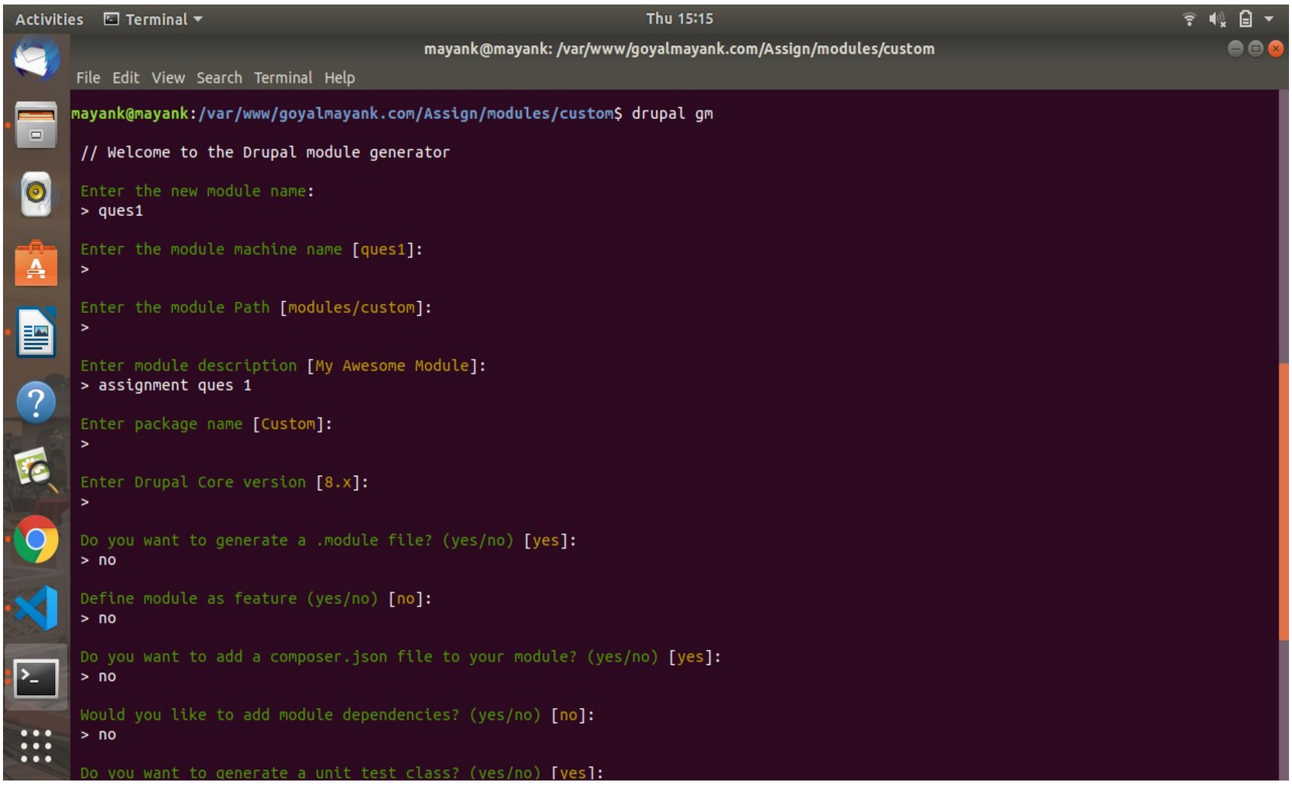
$ drupal


Activities
Terminal
Thu 15:14
mayank@mayank: /var/www/goyalmayank.com/Assign/modules/custom
File Edit View Search Terminal Help

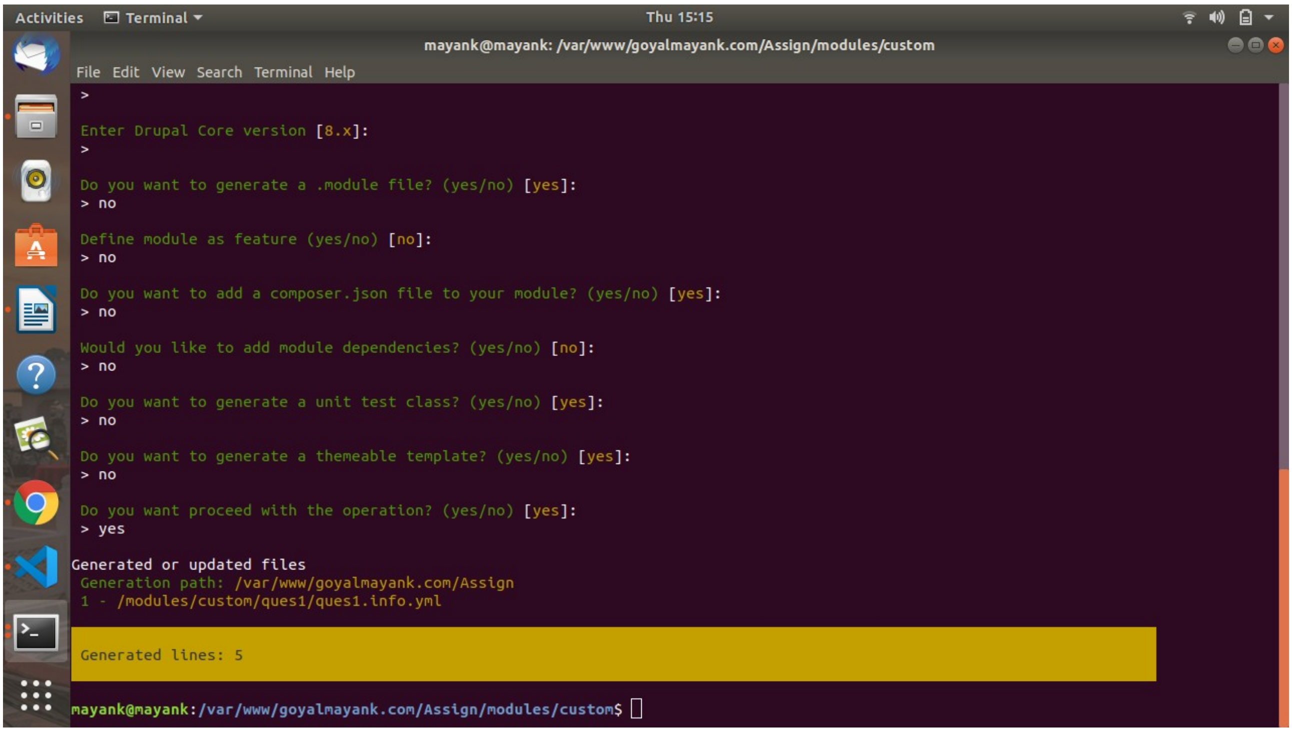
```
mayank@mayank:/var/www/goyalmayank.com/Assign/modules/custom$ drupal
  Drupal Console version 1.9.4

Usage:
  command [options] [arguments]

Options:
  -h, --help              Display this help message
  -q, --quiet             Do not output any message
  -V, --version           Display this application version
      --ansi              Force ANSI output
      --no-ansi           Disable ANSI output
  -n, --no-interaction    Do not ask any interactive question
  -e, --env[=ENV]         The Environment name [default: "prod"]
      --root[=ROOT]       Define the Drupal root to be used in command execution
      --debug             Switches on debug mode
      --learning          Generate a verbose code output
  -c, --generate-chain    Shows command options and arguments as yaml output to be used in chain command
  -i, --generate-inline   Shows command options and arguments as inline command
  -d, --generate-doc      Shows command options and arguments as markdown
  -t, --target[=TARGET]   Site name you want to interact with (for local or remote sites)
  -l, --uri=URI           URI of the Drupal site to use (for multi-site environments or when running on an alternate port)
  -y, --yes               Skip confirmation and proceed
  -v|vv|vvv, --verbose    Increase the verbosity of messages: 1 for normal output, 2 for more verbose output and 3 for debug

Available commands:
  about                                 Displays basic information about Drupal Console project
  chain                                 Chain command execution
  check                                 System requirement checker
  composerize                           Converts Drupal codebase to composer.
  exec                                  Execute an external command.
  help                                  Displays help for a command
  init                                  Copy configuration files.
  list                                  Lists all available commands
  server (serve,rs)                     Runs PHP built-in web server
  shell                                 Open a shell providing an interactive REPL (Read-Eval-Print Loop)
```

step 3: - create a custom module

use command $ drupal gm

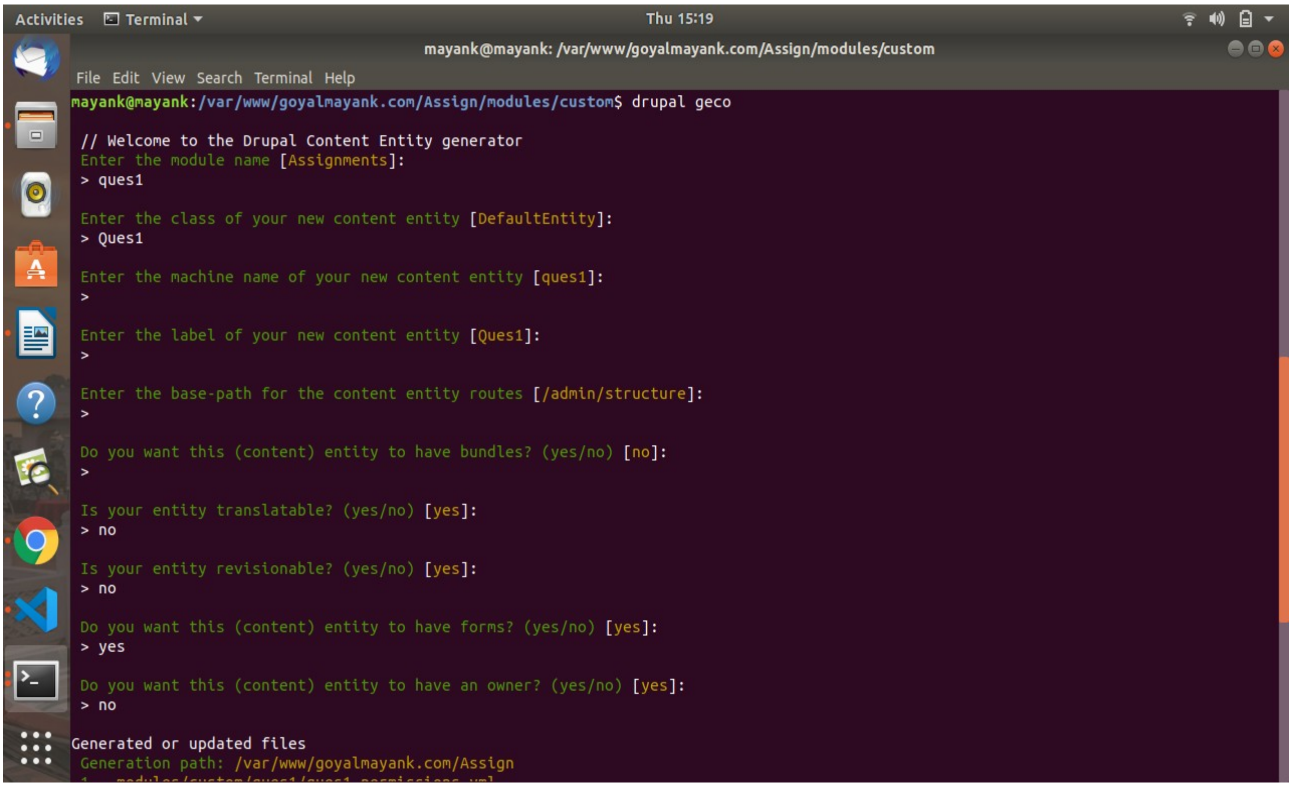

Activities
Terminal
Thu 15:15
mayank@mayank: /var/www/goyalmayank.com/Assign/modules/custom
File Edit View Search Terminal Help

```
mayank@mayank:/var/www/goyalmayank.com/Assign/modules/custom$ drupal gm

 // Welcome to the Drupal module generator

 Enter the new module name:
 > ques1

 Enter the module machine name [ques1]:
 >

 Enter the module Path [modules/custom]:
 >

 Enter module description [My Awesome Module]:
 > assignment ques 1

 Enter package name [Custom]:
 >

 Enter Drupal Core version [8.x]:
 >

 Do you want to generate a .module file? (yes/no) [yes]:
 > no

 Define module as feature (yes/no) [no]:
 > no

 Do you want to add a composer.json file to your module? (yes/no) [yes]:
 > no

 Would you like to add module dependencies? (yes/no) [no]:
 > no

 Do you want to generate a unit test class? (yes/no) [yes]:
```

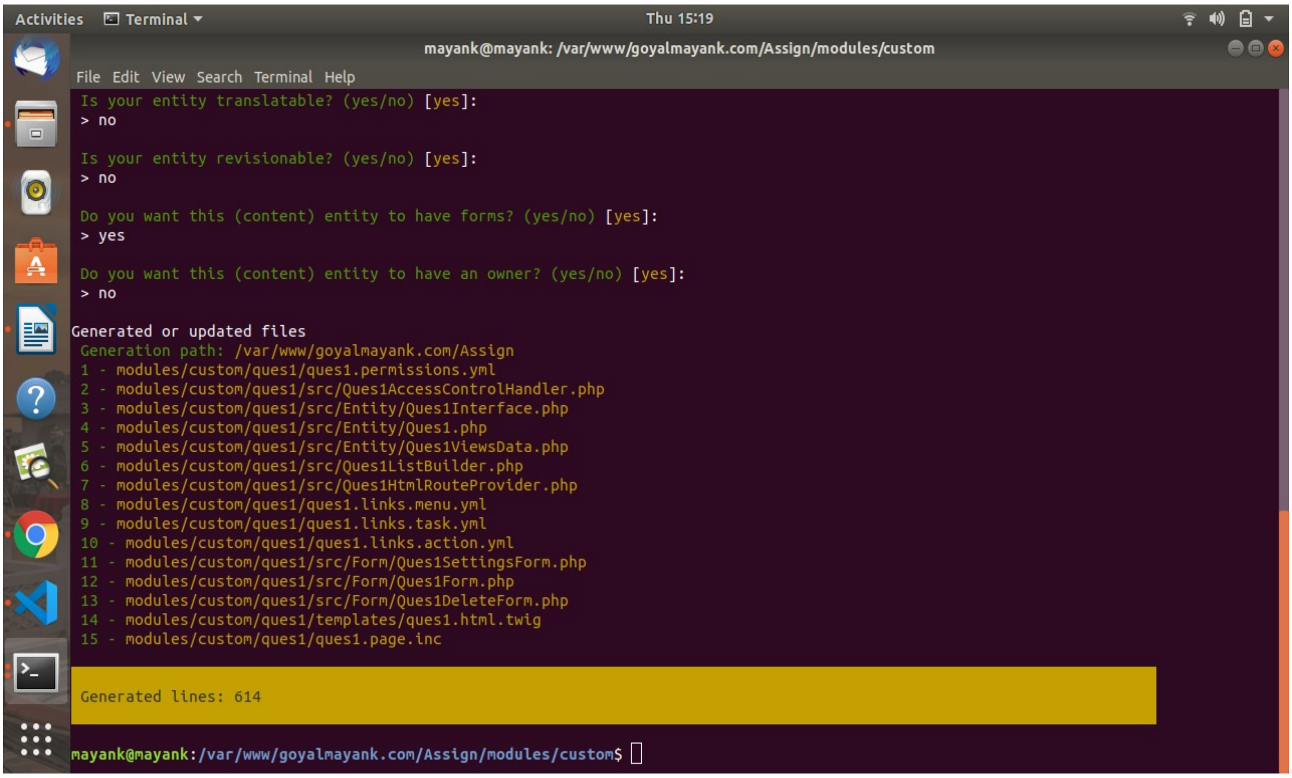

Activities
Terminal
Thu 15:15
mayank@mayank: /var/www/goyalmayank.com/Assign/modules/custom
File Edit View Search Terminal Help

```
 >

 Enter Drupal Core version [8.x]:
 >

 Do you want to generate a .module file? (yes/no) [yes]:
 > no

 Define module as feature (yes/no) [no]:
 > no

 Do you want to add a composer.json file to your module? (yes/no) [yes]:
 > no

 Would you like to add module dependencies? (yes/no) [no]:
 > no

 Do you want to generate a unit test class? (yes/no) [yes]:
 > no

 Do you want to generate a themeable template? (yes/no) [yes]:
 > no

 Do you want proceed with the operation? (yes/no) [yes]:
 > yes

Generated or updated files
 Generation path: /var/www/goyalmayank.com/Assign
 1 - /modules/custom/ques1/ques1.info.yml

 Generated lines: 5

mayank@mayank:/var/www/goyalmayank.com/Assign/modules/custom$
```

step 3:- create custom entity using command $ drupal geco

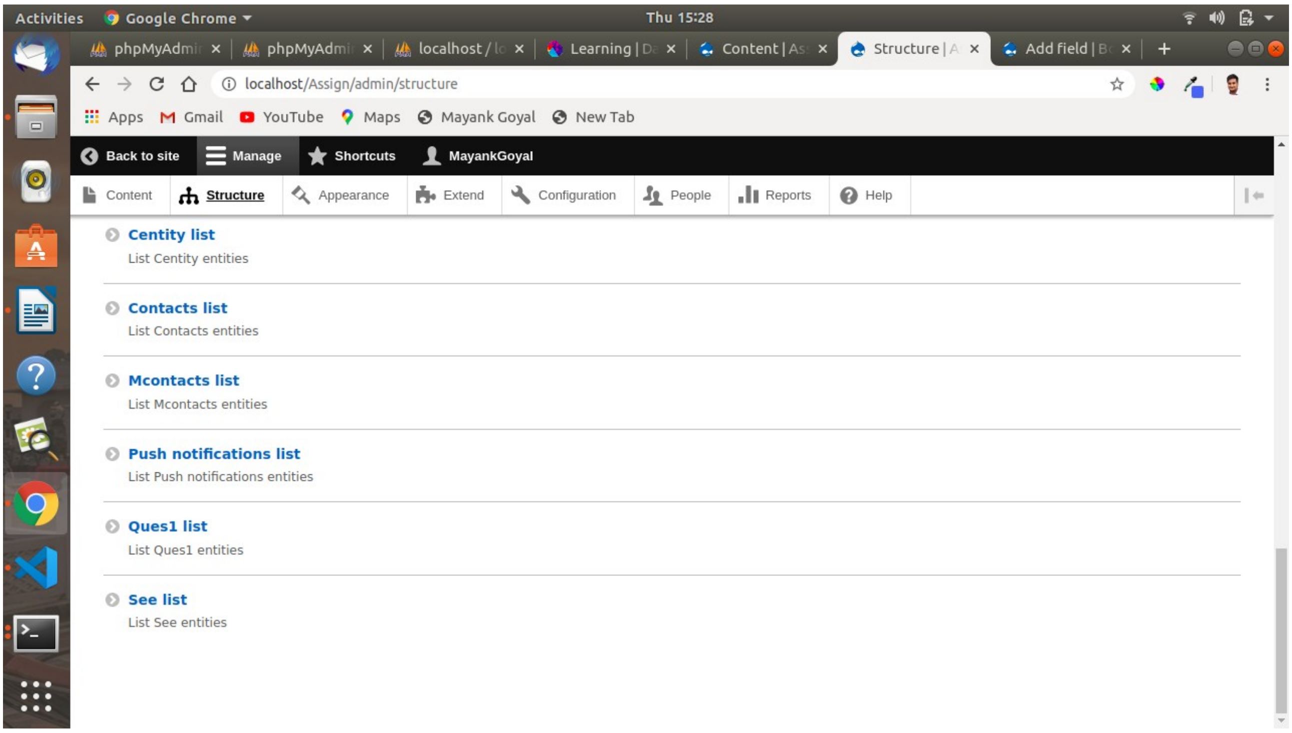

Activities
Terminal
Thu 15:19
mayank@mayank: /var/www/goyalmayank.com/Assign/modules/custom
File Edit View Search Terminal Help

```
mayank@mayank:/var/www/goyalmayank.com/Assign/modules/custom$ drupal geco

 // Welcome to the Drupal Content Entity generator
 Enter the module name [Assignments]:
 > ques1

 Enter the class of your new content entity [DefaultEntity]:
 > Ques1

 Enter the machine name of your new content entity [ques1]:
 >

 Enter the label of your new content entity [Ques1]:
 >

 Enter the base-path for the content entity routes [/admin/structure]:
 >

 Do you want this (content) entity to have bundles? (yes/no) [no]:
 >

 Is your entity translatable? (yes/no) [yes]:
 > no

 Is your entity revisionable? (yes/no) [yes]:
 > no

 Do you want this (content) entity to have forms? (yes/no) [yes]:
 > yes

 Do you want this (content) entity to have an owner? (yes/no) [yes]:
 > no

Generated or updated files
 Generation path: /var/www/goyalmayank.com/Assign
```


Activities
Terminal
Thu 15:19
mayank@mayank: /var/www/goyalmayank.com/Assign/modules/custom
File Edit View Search Terminal Help

```
 Is your entity translatable? (yes/no) [yes]:
 > no

 Is your entity revisionable? (yes/no) [yes]:
 > no

 Do you want this (content) entity to have forms? (yes/no) [yes]:
 > yes

 Do you want this (content) entity to have an owner? (yes/no) [yes]:
 > no

Generated or updated files
 Generation path: /var/www/goyalmayank.com/Assign
 1 - modules/custom/ques1/ques1.permissions.yml
 2 - modules/custom/ques1/src/Ques1AccessControlHandler.php
 3 - modules/custom/ques1/src/Entity/Ques1Interface.php
 4 - modules/custom/ques1/src/Entity/Ques1.php
 5 - modules/custom/ques1/src/Entity/Ques1ViewsData.php
 6 - modules/custom/ques1/src/Ques1ListBuilder.php
 7 - modules/custom/ques1/src/Ques1HtmlRouteProvider.php
 8 - modules/custom/ques1/ques1.links.menu.yml
 9 - modules/custom/ques1/ques1.links.task.yml
 10 - modules/custom/ques1/ques1.links.action.yml
 11 - modules/custom/ques1/src/Form/Ques1SettingsForm.php
 12 - modules/custom/ques1/src/Form/Ques1Form.php
 13 - modules/custom/ques1/src/Form/Ques1DeleteForm.php
 14 - modules/custom/ques1/templates/ques1.html.twig
 15 - modules/custom/ques1/ques1.page.inc

 Generated lines: 614

mayank@mayank:/var/www/goyalmayank.com/Assign/modules/custom$ 
```

now go to your website , install the module and clear the cache



Centity list
List Centity entities

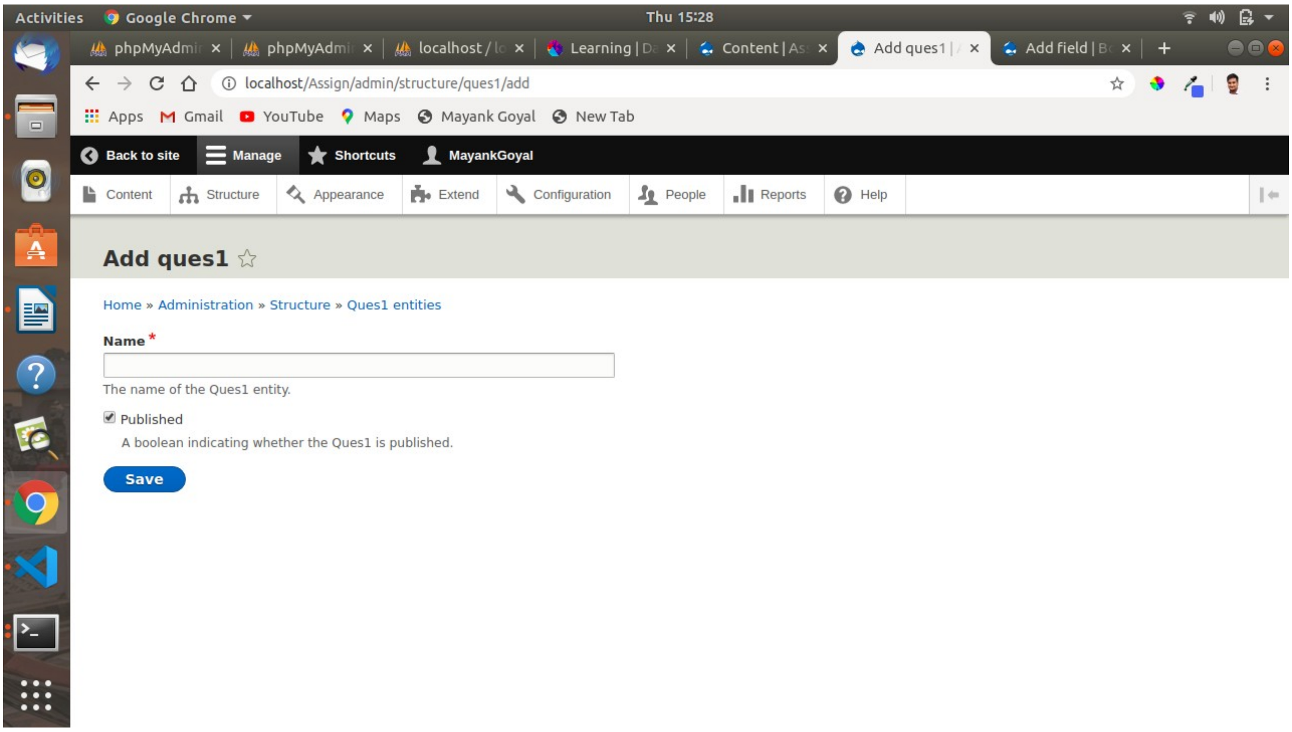
Contacts list
List Contacts entities

Mcontacts list
List Mcontacts entities

Push notifications list
List Push notifications entities

Ques1 list
List Ques1 entities

See list
List See entities



Add ques1 ☆

Home » Administration » Structure » Ques1 entities

Name *

The name of the Ques1 entity.

☑ Published

A boolean indicating whether the Ques1 is published.

Save

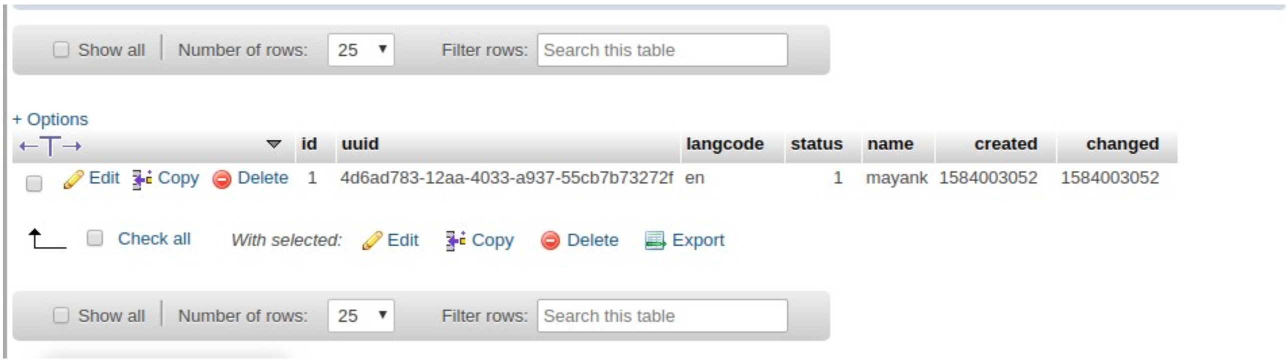
To add the fields email and contact go to your module

goto src >> entity

now in the php file do the following changes :-

```
<?php

namespace Drupal\seecont\Entity;

use Drupal\Core\Field\BaseFieldDefinition;
use Drupal\Core\Entity\ContentEntityBase;
use Drupal\Core\Entity\EntityChangedTrait;
use Drupal\Core\Entity\EntityPublishedTrait;
use Drupal\Core\Entity\EntityTypeInterface;

/**
 * Defines the See entity.
 *
 * @ingroup seecont
 *
 * @ContentEntityType(
 *   id = "see",
 *   label = @Translation("See"),
 *   handlers = {
 *     "view_builder" = "Drupal\Core\Entity\EntityViewBuilder",
 *     "list_builder" = "Drupal\seecont\SeeListBuilder",
 *     "views_data" = "Drupal\seecont\Entity\SeeViewsData",
 *
 *     "form" = {
 *       "default" = "Drupal\seecont\Form\SeeForm",
 *       "add" = "Drupal\seecont\Form\SeeForm",
 *       "edit" = "Drupal\seecont\Form\SeeForm",
 *       "delete" = "Drupal\seecont\Form\SeeDeleteForm",
 *     },
 *     "route_provider" = {
 *       "html" = "Drupal\seecont\SeeHtmlRouteProvider",
 *     },
 *     "access" = "Drupal\seecont\SeeAccessControlHandler",
 *   },
 *   base_table = "see",
 *   translatable = FALSE,
 *   admin_permission = "administer see entities",
 *   entity_keys = {
 *     "id" = "id",
 *     "label" = "name",
 *     "uuid" = "uuid",
 *     "langcode" = "langcode",
 *     "published" = "status",
 *   },
 *   links = {
 *     "canonical" = "/admin/structure/see/{see}",
 *     "add-form" = "/admin/structure/see/add",
 *     "edit-form" = "/admin/structure/see/{see}/edit",
 *     "delete-form" = "/admin/structure/see/{see}/delete",
 *     "collection" = "/admin/structure/see",
 *   },
 *   field_ui_base_route = "see.settings"
 * )
 */
class See extends ContentEntityBase implements SeeInterface {

  use EntityChangedTrait;
  use EntityPublishedTrait;

  /**
   * {@inheritdoc}
   */
  public function getName() {
    return $this->get('name')->value;
  }

  /**
   * {@inheritdoc}
   */
  public function setName($name) {
    $this->set('name', $name);
    return $this;
  }

  /**
   * {@inheritdoc}
   */
  public function getCreatedTime() {
    return $this->get('created')->value;
  }

  /**
   * {@inheritdoc}
   */
  public function setCreatedTime($timestamp) {
    $this->set('created', $timestamp);
    return $this;
  }

  /**
   * {@inheritdoc}
   */
  public static function baseFieldDefinitions(EntityTypeInterface $entity_type) {
    $fields = parent::baseFieldDefinitions($entity_type);

    // Add the published field.
    $fields += static::publishedBaseFieldDefinitions($entity_type);
  // name entry starts
    $fields['name'] = BaseFieldDefinition::create('string')
      ->setLabel(t('Name'))
      ->setDescription(t('The name of the See entity.'))
      ->setSettings([
        'max_length' => 50,
        'text_processing' => 0,
      ])
      ->setDefaultValue('')
      ->setDisplayOptions('view', [
        'label' => 'above',
        'type' => 'string',
        'weight' => -4,
      ])
      ->setDisplayOptions('form', [
        'type' => 'string_textfield',
        'weight' => -4,
      ])
      ->setDisplayConfigurable('form', TRUE)
      ->setDisplayConfigurable('view', TRUE)
      ->setRequired(TRUE);
      // entering email

      $fields['Email'] = BaseFieldDefinition::create('string')
      ->setLabel(t('email'))
      ->setDescription(t('The e-mail of the See entity.'))
      ->setSettings([
        'max_length' => 50,
        'text_processing' => 0,
      ])
      ->setDefaultValue('')
      ->setDisplayOptions('view', [
        'label' => 'above',
        'type' => 'string',
        'weight' => -4,
      ])
      ->setDisplayOptions('form', [
        'type' => 'string_textfield',
        'weight' => -4,
      ])
      ->setDisplayConfigurable('form', TRUE)
      ->setDisplayConfigurable('view', TRUE)
      ->setRequired(TRUE);
      // contact
      $fields['contact'] = BaseFieldDefinition::create('integer')
      ->setLabel(t('contact'))
      ->setDescription(t('The contact of the See entity.'))
      ->setSettings([
        'max_length' => 10,
        'text_processing' => 0,
      ])
      ->setDefaultValue('')
      ->setDisplayOptions('view', [
        'label' => 'above',
        'type' => 'integer',
        'weight' => -4,
      ])
      ->setDisplayOptions('form', [
        'type' => 'string_textfield',
        'weight' => -4,
      ])
      ->setDisplayConfigurable('form', TRUE)
      ->setDisplayConfigurable('view', TRUE)
      ->setRequired(TRUE);


    $fields['status']->setDescription(t('A boolean indicating whether the See is published.'))
      ->setDisplayOptions('form', [
        'type' => 'boolean_checkbox',
        'weight' => -3,
      ]);

    $fields['created'] = BaseFieldDefinition::create('created')
      ->setLabel(t('Created'))
      ->setDescription(t('The time that the entity was created.'));

    $fields['changed'] = BaseFieldDefinition::create('changed')
      ->setLabel(t('Changed'))
      ->setDescription(t('The time that the entity was last edited.'));

    return $fields;
  }

}
```

now go to your website , clear cache and check the form :-



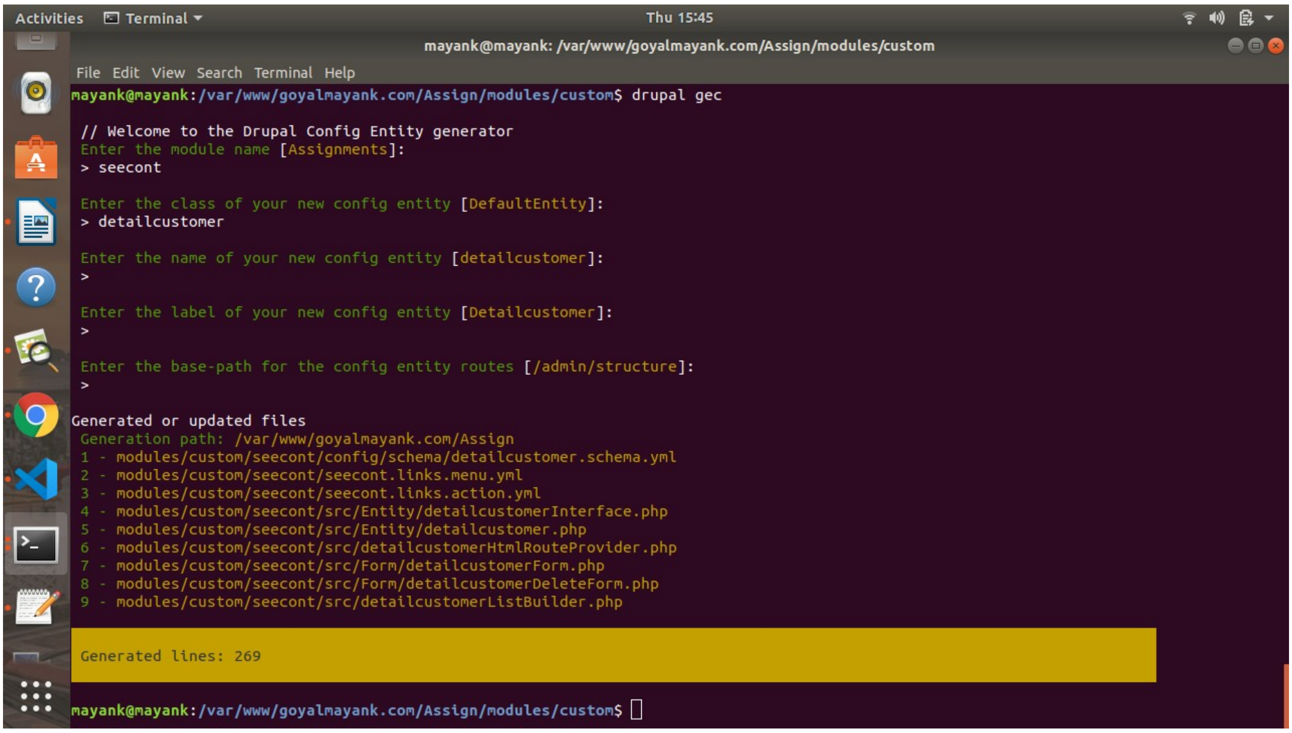
id	uuid	langcode	status	name	created	changed
1	4d6ad783-12aa-4033-a937-55cb7b73272f	en	1	mayank	1584003052	1584003052

Ques 2 :- Create a Config Entity to store diff time slots for customers.

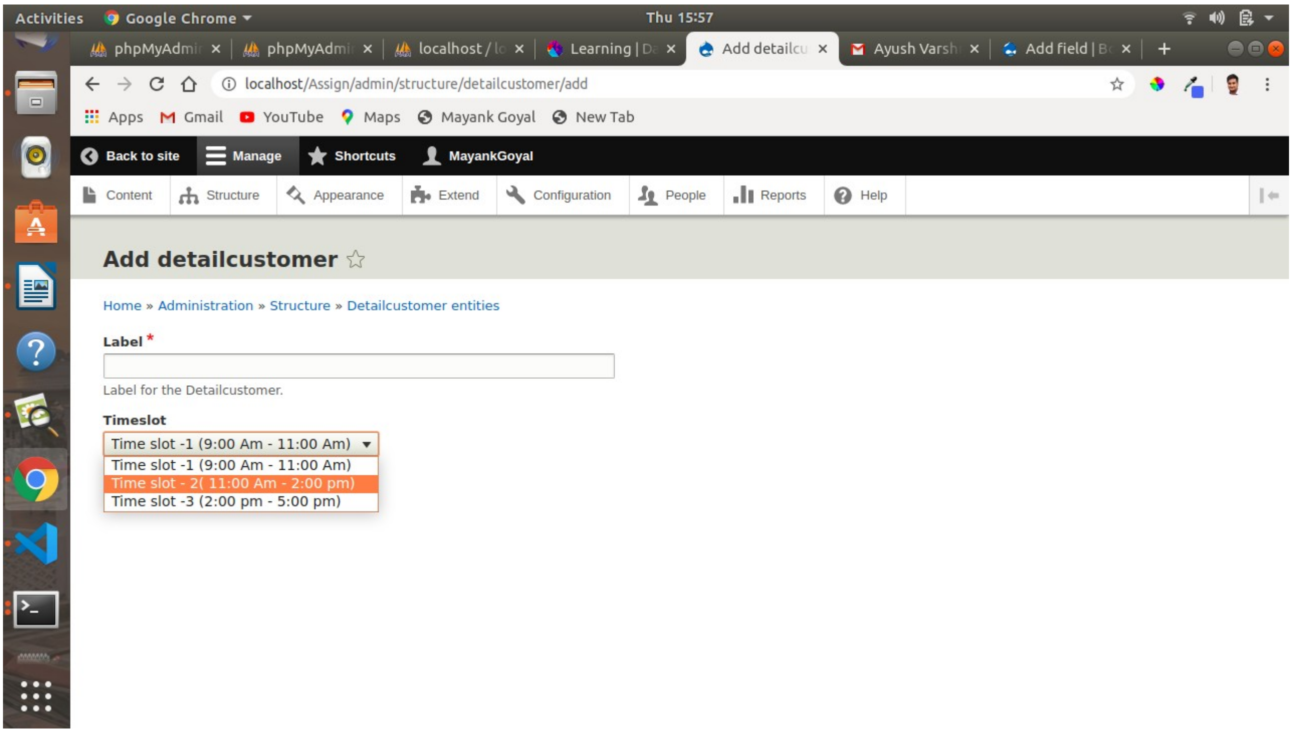
Soln:-

step 1:- run the following command

$ drupal gec


Activities
Terminal
Thu 15:45
mayank@mayank: /var/www/goyalmayank.com/Assign/modules/custom
File Edit View Search Terminal Help

```
mayank@mayank:/var/www/goyalmayank.com/Assign/modules/custom$ drupal gec

 // Welcome to the Drupal Config Entity generator
 Enter the module name [Assignments]:
 > seecont

 Enter the class of your new config entity [DefaultEntity]:
 > detailcustomer

 Enter the name of your new config entity [detailcustomer]:
 >

 Enter the label of your new config entity [Detailcustomer]:
 >

 Enter the base-path for the config entity routes [/admin/structure]:
 >

Generated or updated files
 Generation path: /var/www/goyalmayank.com/Assign
 1 - modules/custom/seecont/config/schema/detailcustomer.schema.yml
 2 - modules/custom/seecont/seecont.links.menu.yml
 3 - modules/custom/seecont/seecont.links.action.yml
 4 - modules/custom/seecont/src/Entity/detailcustomerInterface.php
 5 - modules/custom/seecont/src/Entity/detailcustomer.php
 6 - modules/custom/seecont/src/detailcustomerHtmlRouteProvider.php
 7 - modules/custom/seecont/src/Form/detailcustomerForm.php
 8 - modules/custom/seecont/src/Form/detailcustomerDeleteForm.php
 9 - modules/custom/seecont/src/detailcustomerListBuilder.php


 Generated lines: 269

mayank@mayank:/var/www/goyalmayank.com/Assign/modules/custom$
```



Add detailcustomer ☆

Home » Administration » Structure » Detailcustomer entities

Label *

Label for the Detailcustomer.

Timeslot

Time slot -1 (9:00 Am - 11:00 Am) ▾
Time slot -1 (9:00 Am - 11:00 Am)
Time slot - 2(11:00 Am - 2:00 pm)
Time slot -3 (2:00 pm - 5:00 pm)

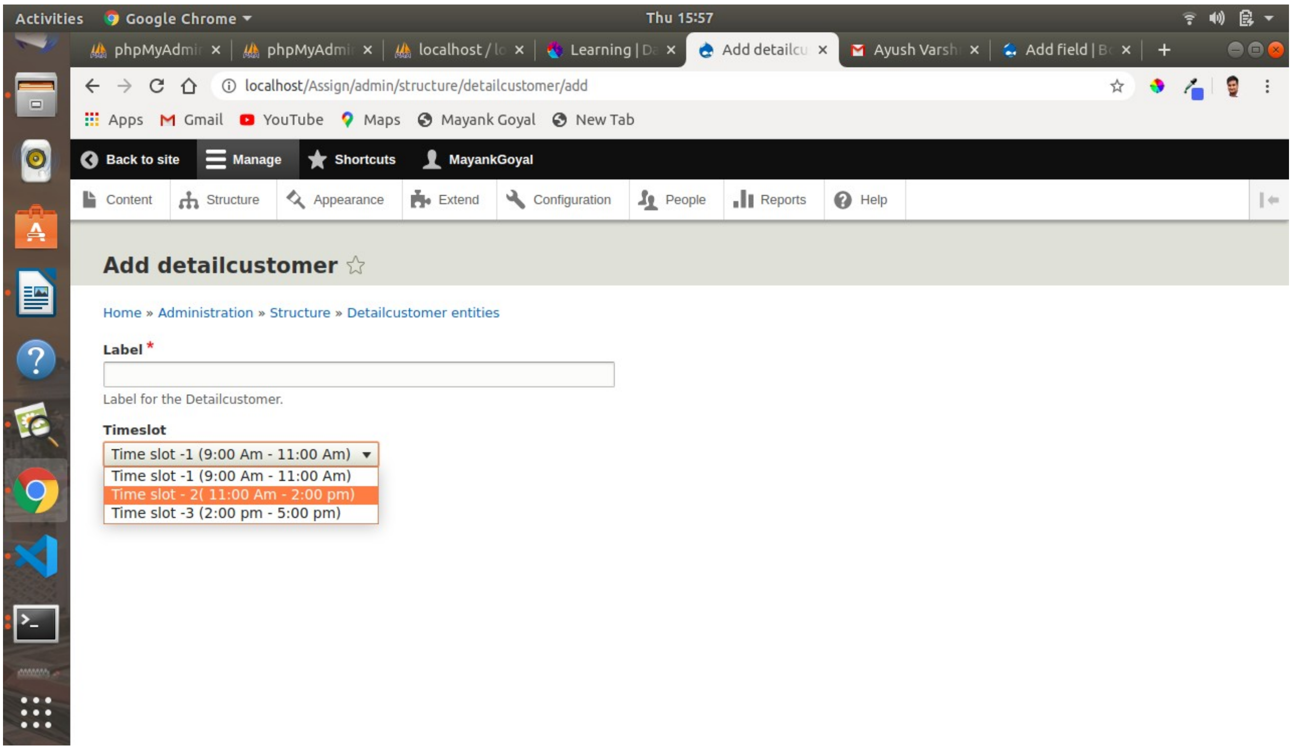
step 2:- go to module >> custom >> your module name >> src
step3 :- Open form folder and inside detailcustomerForm.php do the following changes :-

```
<?php

namespace Drupal\seecont\Form;

use Drupal\Core\Entity\EntityForm;
use Drupal\Core\Form\FormStateInterface;

/**
 * Class detailcustomerForm.
 */
class detailcustomerForm extends EntityForm {

  /**
   * {@inheritdoc}
   */
  public function form(array $form, FormStateInterface $form_state) {
    $form = parent::form($form, $form_state);

    $detailcustomer = $this->entity;
    $form['label'] = [
      '#type' => 'textfield',
      '#title' => $this->t('Label'),
      '#maxlength' => 255,
      '#default_value' => $detailcustomer->label(),
      '#description' => $this->t("Label for the Detailcustomer."),
      '#required' => TRUE,
    ];

    $form['id'] = [
      '#type' => 'machine_name',
      '#default_value' => $detailcustomer->id(),
      '#machine_name' => [
        'exists' => '\Drupal\seecont\Entity\detailcustomer::load',
      ],
      '#disabled' => !$detailcustomer->isNew(),
    ];
    $form['timeslot'] = [
      '#type' => 'select',
      '#options'=>[1=>'Time slot -1 (9:00 Am - 11:00 Am)' ,2=>'Time slot - 2( 11:00 Am -
2:00 pm)',3=>'Time slot -3 (2:00 pm - 5:00 pm)' ],
      '#title' => 'Timeslot',
      // '#default_value' => $detailcustomer->timeslot(),
      // '#machine_name' => [
      //   'exists' => '\Drupal\seecont\Entity\detailcustomer::load',
      // ],
      // '#disabled' => !$detailcustomer->isNew(),
    ];

    /* You will need additional form elements for your custom properties. */

    return $form;
  }

  /**
   * {@inheritdoc}
   */
  public function save(array $form, FormStateInterface $form_state) {
    $detailcustomer = $this->entity;
    $status = $detailcustomer->save();

    switch ($status) {
      case SAVED_NEW:
        $this->messenger()->addMessage($this->t('Created the %label Detailcustomer.', [
          '%label' => $detailcustomer->label(),
        ]));
        break;

      default:
        $this->messenger()->addMessage($this->t('Saved the %label Detailcustomer.', [
          '%label' => $detailcustomer->label(),
        ]));
    }
    $form_state->setRedirectUrl($detailcustomer->toUrl('collection'));
  }

}
```

Step :- now go to config >> schema

Step2: in file schema.yml do the following changes:-

```
seecont.detailcustomer.*:
  type: config_entity
  label: 'Detailcustomer config'
  mapping:
    id:
      type: string
      label: 'ID'
    label:
      type: label
      label: 'Label'
    uuid:
      type: string
    timeslot:
      type: string
      label: 'Timeslot'
```



Add detailcustomer ☆

Home » Administration » Structure » Detailcustomer entities

Label *

Label for the Detailcustomer.

Timeslot

Time slot -1 (9:00 Am - 11:00 Am) ▾

Time slot -1 (9:00 Am - 11:00 Am)
Time slot - 2(11:00 Am - 2:00 pm)
Time slot -3 (2:00 pm - 5:00 pm)