1. **Write the filename where we use and write hooks.**

   All hooks almost always go in the .module file. The .module file is loaded on every page load.

   Module -> mymodule -> mymodule.module file

2. **Write any 10 hooks name and their definition which is used by drupal 8.**

   # hook_cron -> Perform periodic actions.

   # hook_entity_insert -> Respond to creation of a new entity.

   # hook_form_alter -> Perform alterations before a form is rendered.

   # hook_install -> Perform setup tasks when the module is installed.

   # hook_mail -> Prepares a message based on parameters;

   # hook_preprocess -> Preprocess theme variables for templates.

   # hook_schema ->Define the current version of the database schema.

   # hook_theme ->Register a module or theme's theme implementations.

   # hook_uninstall -> Remove any information that the module sets.

   # hook_update_N ->Perform a single update between minor versions.

   # hook_user_login ->The user just logged in.

3.

Create a custom module and use any 3 hooks and their implementation.

```
function hookModule_form_alter(&$form, \Drupal\Core\Form\FormStateInterface $form_state, $form_id)
  {

      if($form_id == 'comment_comment_form')
      {
        $form['actions']['submit']['#value'] = t('Comment');
        $form['actions']['preview']['#value'] = t('PREVIEW');
      }
  }
```

```
 function hookModule_page_top(array &$page_top)
 {
   $page_top['hookModule'] =
   [
     '#markup' => '<div role="contentinfo" aria-label="Status message" class="messages">
                  <h2>Status message</h2>This is Top of the Page</div>',
   ];
 }
```

```
function hookModule_page_bottom(array &$page_bottom)
{
  $page_bottom['hookmodule'] = [
    '#markup' => '<div role="contentinfo" aria-label="Status message" class="messages">
                  <h2>Status message</h2>This is Bottom of the Page</div>',
  ];
}
```

```php
function hookModule_user_format_name_alter(&$name, $account)
{
  // Display the user's uid instead of name.
  if ($account
    ->id())
  {
    $name = t(strtoupper($name).' : @uid', [
      '@uid' => $account
        ->id(),
    ]);
  }
}
```

```php
function hookModule_user_login($account)
  {
      \Drupal::logger('hookModule')
       ->info('Logged In : '.$account->getDisplayName());
  }


function hookModule_user_logout($account)
{
      \Drupal::logger('hookModule')
      ->info('User LoggedOut : '.$account->getDisplayName());

      \Drupal::database()
       ->insert('logouts')
       ->fields([
       'uid' => $account
        ->id(),
    ])
  ->execute();
}
```

4.What is the use of .theme file.

5.What is theme() ?

`Drupal.theme` is the client-side counterpart of the server-side `theme()`-function. All of the server-side `theme()` function has been removed in Drupal 8. The client-side function however remains part of the Javascript API. All markup (defined in Javascript) should go through these theme functions.

The beauty of this system is that it allows the reuse of theme functions as templates and the override of these functions in themes. A module can define a theme function; a theme that wants to change the markup can override the function to define its own template.

The `Drupal.theme` function has been simplified. Previously it was possible to declare functions in `Drupal.theme.prototype`, where workaround code made `Drupal.theme()` behave as intended. Instead these Javascript theme functions are now using `Drupal.theme` directly.

4. **Why we use preprocess functions and how many type of preprocess function in drupal 8?**

   Preprocess functions allow **Drupal** themes to manipulate the variables that are used in Twig template files by using PHP **functions** to **preprocess** data before it is exposed to each template. All of the dynamic content available to theme developers within a Twig template file is exposed through a **preprocess function**

● **template_preprocess(&$variables, $hook):** Creates a default set of variables for all theme hooks with template implementations. Provided by Drupal Core.

● **template_preprocess_HOOK(&$variables):** Should be implemented by the module that registers the theme hook, to set up default variables.

● **MODULE_preprocess(&$variables, $hook):** hook_preprocess() is invoked on all implementing modules.

● **MODULE_preprocess_HOOK(&$variables):** hook_preprocess_HOOK() is invoked on all implementing modules, so that modules that didn't define the theme hook can alter the variables.

● **ENGINE_engine_preprocess(&$variables, $hook):** Allows the theme engine to set necessary variables for all theme hooks with template implementations.

● **ENGINE_engine_preprocess_HOOK(&$variables):** Allows the theme engine to set necessary variables for the particular theme hook.

● **THEME_preprocess(&$variables, $hook):** Allows the theme to set necessary variable for all theme hooks with template implementations.

● **THEME_preprocess_HOOK(&$variables):** Allows the theme to set necessary variables specific to the particular theme hook.

7. In which file we write the preprocess functions?

In .theme file

8. What is template engine in drupal 8?

Drupal 8 uses the templating engine Twig. Twig offers developers a fast, secure, and flexible method for building templates for Drupal 8 sites. Twig also offers substantial usability improvements over PHPTemplate, and does not require front-end developers to know PHP to build and manipulate Drupal 8 themes.

9.
Add one text field programatically at the last in any one content type with your name and save your name and add a validation for alphabet.

File uploaded.