

1 Create a WordPress plugin which contains the following functionality

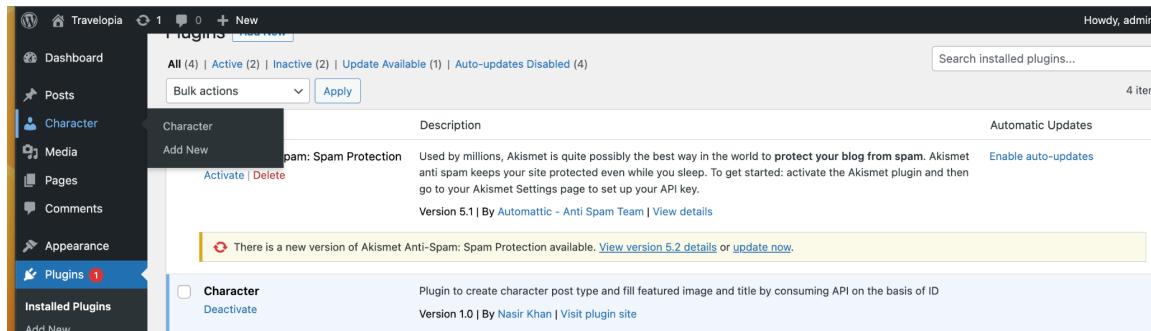


The screenshot shows the WordPress admin interface under the 'Plugins' section. A new plugin, 'Character', has been installed and is listed in the 'Installed Plugins' section. The plugin's description is: 'Plugin to create character post type and fill featured image and title by consuming API on the basis of ID'. It is version 1.0 and was created by Nasir Khan. The code editor on the right displays the PHP code for the plugin, specifically the 'character.php' file. The code includes logic for adding a featured image to posts and handling folder permissions.

```
126 }  
127 // Add Featured Image to Post  
128 $image_url = preg_replace('/\?.*/', '', $url); // removing query string from url & Define the i  
129 $image_name = basename($image_url);  
130 $upload_dir = wp_upload_dir(); // Set upload folder  
131 $image_data = file_get_contents($url); // Get image data  
132 $unique_file_name = wp_unique_filename( $upload_dir['path'], $image_name ); // Generate unique name  
133 $filename = basename( $unique_file_name ); // Create image file name  
134  
135 // Check folder permission and define file location  
136 if( wp_mkdir_p( $upload_dir['path'] ) ) {  
137     $file = $upload_dir['path'] . '/' . $filename;  
138 } else {  
139     $file = $upload_dir['basedir'] . '/' . $filename;  
140 }  
141 }
```

Plugin created to create custom post type Character.

2 Create a post type “Character”



Custom post type created with the help of plugin

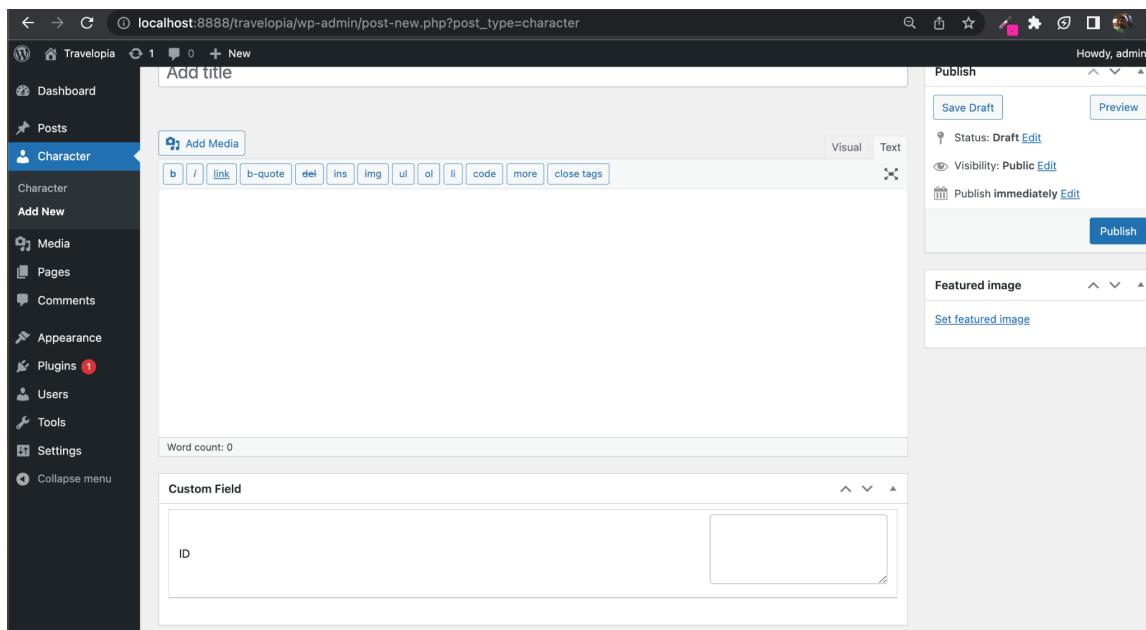
```

21
22
23
24     /* Author : Nasir Khan
25      * Description : Create character custom post type
26      *
27      */
28
29     function character_post_type() {
30         register_post_type( 'character',
31             array(
32                 'labels' => array(
33                     'name' => __( 'Character' ),
34                     'singular_name' => __( 'character' )
35                 ),
36                 'public' => true,
37                 'show_in_rest' => true,
38                 'supports' => array('title', 'editor', 'thumbnail', 'custom-fields'),
39                 'has_archive' => true,
40                 'menu_position' => 5,
41                 'menu_icon' => 'dashicons-admin-users',
42             )
43         );
44     }
45     add_action( 'init', 'character_post_type' );
46
47
48     /**

```

For this used register_post_type function in wordpress and pass the array and assign it to init hook

3 Create a custom field: ID for the post type Character (without a plugin)



```

    character
      character.php
    classic-editor
      hello.php
      index.php
  themes
    MyTheme
      img
      js
        custom.js
      # custom.css
      footer.php
      functions.php
      header.php
      index.php
      page.php
      screenshot.png
      single.php
      # style.css
      index.php
  > OUTLINE

```

```

28  /* Author : Nasir Khan
29  *   Description : Function to create custom field under post
30  */
31 */
32
33
34 function character_add_meta_box( $post ){
35   add_meta_box( 'character_meta_box', 'Custom Field', 'character_build_meta_box', 'character', 'normal',
36 }
add_action( 'add_meta_boxes_character', 'character_add_meta_box' );
37
38
39 function character_build_meta_box( $post ){
40   wp_nonce_field( basename( __FILE__ ), 'character_meta_box_nonce' );
41   $id_custom_field = get_post_meta( $post->ID, 'ID', true );
42   <div id="postcustom" class="postbox" >
43     <div class="postbox-header"><h2 class="hndle ui-sortable-handle">ID</h2>
44     <div class='inside'>
45       <textarea id="metavalue" name="id_custom_field" rows="2" cols="25"><?php echo $id_custom_field; ?></tex
46     </div>
47   </div>
48 </div>
49 </?php >
50
51

```

```

    wp-admin
  wp-content
    plugins
      akismet
    character
      character.php
    classic-editor
      hello.php
      index.php
  themes
    MyTheme
      img
      js
        custom.js
      # custom.css
      footer.php
      functions.php
      header.php
      index.php
      page.php
      screenshot.png
      single.php
      # style.css
      index.php
  > OUTLINE

```

```

45   <!-- -->
46   <textarea id="metavalue" name="id_custom_field" rows="2" cols="25"><?php echo $id_custom_field; ?></t
47   </div>
48   </div>
49   </div>
50
51
52 function character_save_meta_box_data( $post_id ){
53   if ( !isset( $_POST['character_meta_box_nonce'] ) || !wp_verify_nonce( $_POST['character_meta_box_nonce'],
54     return;
55   }
56
57   if ( isset( $_REQUEST['id_custom_field'] ) ) {
58     update_post_meta( $post_id, 'ID', sanitize_text_field( $_POST['id_custom_field'] ) );
59   }
60
61 add_action( 'save_post_character', 'character_save_meta_box_data', 10, 2 );

```

Saving custom field value with help update_post_meta using save_post_{custom_post_type} hook

4 When a Character post is created or updated, update the post meta

When the character post is created or update the post meta is updated with the help of update_post_meta() function and using save_post hook

- 5a Once the post is updated, run an API call to this API:
<https://thronesapi.com/api/v2/Characters/<id>> with the ID being the one the user entered in the post

```

> images
> js
> includes
> languages
public
> css
> images
js
character.php
classic-editor
hello.php
index.php
themes
MyTheme
> img
js
custom.js
# custom.css
# footer.php
functions.php
header.php
index.php
page.php
# screenshot.png
single.php
# style.css
index.php
upgrade
uploads
index.php
LINE
ELINE
    74
    75 /* Author : Nasir Khan
    76 * Description : Get remote date from api
    77 *
    78 */
    79
    80 function get_remote_value_from_api($post_id){
    81
    82     if ( get_post_type($post_id) == 'character' ) {
    83
    84         $character_id = get_post_meta($post_id, 'ID', true);
    85         if( !isset($character_id) || $character_id != '' ){
    86
    87             $request = wp_remote_get( 'https://thronesapi.com/api/v2/Characters/'.$character_id );
    88             if( is_wp_error( $request ) ){
    89                 return false;
    90             }
    91
    92             // To get data from API
    93             $body = wp_remote_retrieve_body( $request );
    94             $data = json_decode( $body );
    95             $imageUrl = $data->imageUrl; // Featured Image Url
    96
    97             // Update character post type
    98             $character_post_update = array(
    99                 'ID'          => $post_id,
   100                 'post_title'  => $data->fullName,
   101             );
   102
   103             // Function to set featured image in post through Api image Url
   104             set_featured_image_from_api_image_url($imageUrl, $post_id);
   105
   106             remove_action('save_post', 'character_update'); // To avoid infinite loop as we are using wp_update_post with save_post hook
   107             // Update the specified character post into the database
   108             wp_update_post( $character_post_update );
   109             add_action('save_post', 'character_update');
   110

```

wp_remote_post and **wp_remote_retrieve_body** function is used to get json response from API and **json_decode** is used to decode json to php object

- 5b Update the post's title to be the `fullName` field of the character from the API

```

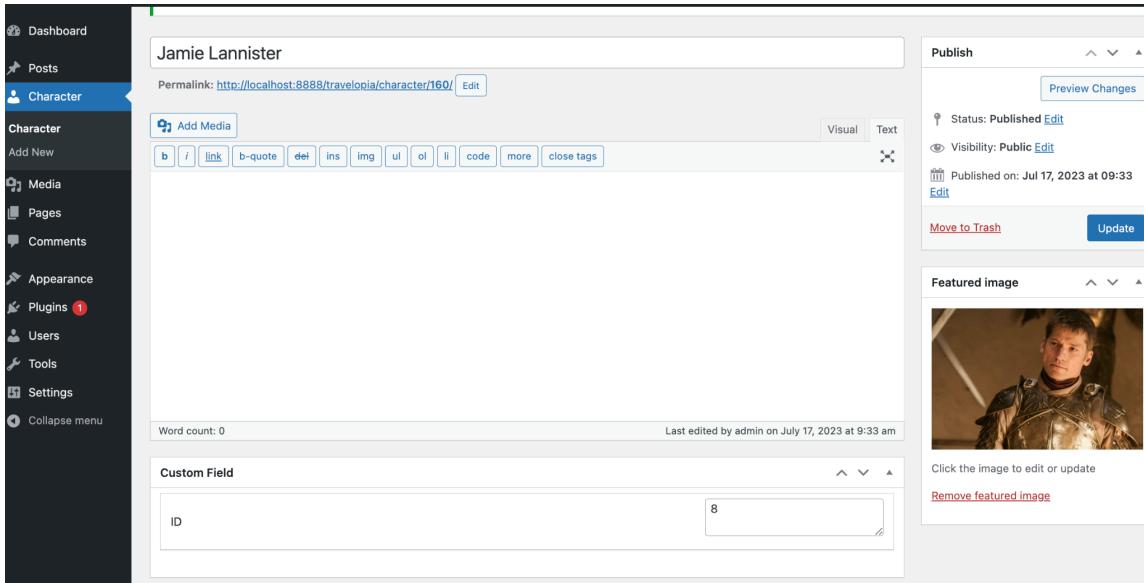
hello.php
index.php
themes
MyTheme
> img
js
custom.js
# custom.css
# footer.php
functions.php
header.php
index.php
page.php
# screenshot.png
single.php
# style.css
index.php
index.php
    90
    91
    92
    93
    94
    95
    96
    97
    98
    99
    100
    101
    102
    103
    104
    105
    106
    107
    108
    109
    110
        }
        // To get data from API
        $body = wp_remote_retrieve_body( $request );
        $data = json_decode( $body );
        $imageUrl = $data->imageUrl; // Featured Image Url

        // Update character post type
        $character_post_update = array(
            'ID'          => $post_id,
            'post_title'  => $data->fullName,
        );

        // Function to set featured image in post through Api image Url
        set_featured_image_from_api_image_url($imageUrl, $post_id);

        remove_action('save_post', 'character_update'); // To avoid infinite loop as we are using wp_update_post with save_post hook
        // Update the specified character post into the database
        wp_update_post( $character_post_update );

```



`wp_update_post` is used to update custom post type character title with fullname

5c Update the post's featured image to be the image from the API `imageUrl`

```

wp-content > plugins > character > character.php
  117  /* Author : Nasir Khan
  118  * Description : Set the featured image from api url
  119  *
  120  */
  121  function set_featured_image_from_api_image_url($url, $post_id){
  122      if ( ! filter_var($url, FILTER_VALIDATE_URL) || empty($post_id) ) {
  123          return;
  124      }
  125
  126      // Add Featured Image to Post
  127      $image_url      = preg_replace('/\?.*/', '', $url); // removing query string from url & Define the image URL here
  128      $image_name     = basename($image_url);
  129      $upload_dir     = wp_upload_dir(); // Set upload folder
  130      $image_data     = file_get_contents($url); // Get image data
  131      $unique_file_name = wp_unique_filename( $upload_dir['path'], $image_name ); // Generate unique name
  132      $filename        = basename( $unique_file_name ); // Create image file name
  133
  134      // Check folder permission and define file location
  135      if( wp_mkdir_p( $upload_dir['path'] ) ) {
  136          $file = $upload_dir['path'] . '/' . $filename;
  137      } else {
  138          $file = $upload_dir['basedir'] . '/' . $filename;
  139      }
  140
  141      // Create the image file on the server
  142      file_put_contents( $file, $image_data );
  143
  144      // Check image file type
  145      $wp_filetype = wp_check_filetype( $filename, null );
  146
  147      // Set attachment data
  148      $attachment = array(
  149          'post_mime_type' => $wp_filetype['type'],
  150          'post_title'     => sanitize_file_name( $filename ),
  151          ...
  152      )
  
```

```
    }
    // Create the image file on the server
    file_put_contents( $file, $image_data );
    // Check image file type
    $wp_filetype = wp_check_filetype( $filename, null );
    // Set attachment data
    $attachment = array(
        'post_mime_type' => $wp_filetype['type'],
        'post_title' => sanitize_file_name( $filename ),
        'post_content' => '',
        'post_status' => 'inherit'
    );
    // Create the attachment
    $attach_id = wp_insert_attachment( $attachment, $file, $post_id );
    // Include image.php
    require_once(ABSPATH . 'wp-admin/includes/image.php');
    // Define attachment metadata
    $attach_data = wp_generate_attachment_metadata( $attach_id, $file );
    // Assign metadata to attachment
    wp_update_attachment_metadata( $attach_id, $attach_data );
    // And finally assign featured image to post
    set_post_thumbnail( $post_id, $attach_id );
}
/* Author : Nasir Khan
 * Description : Get character post type update
 */

```

First uploaded image in media library through code using `file_put_content()` and created the attachment.

`Set_post_thumbnail` function is used to insert featured image from url and for that we need attachment id so `wp_insert_attachment`

The image displays two screenshots of a WordPress-like CMS interface.

Screenshot 1: Character Post Editor

This screenshot shows the post editor for a character named "Khal Drogo". The left sidebar is a dark navigation menu with the following items:

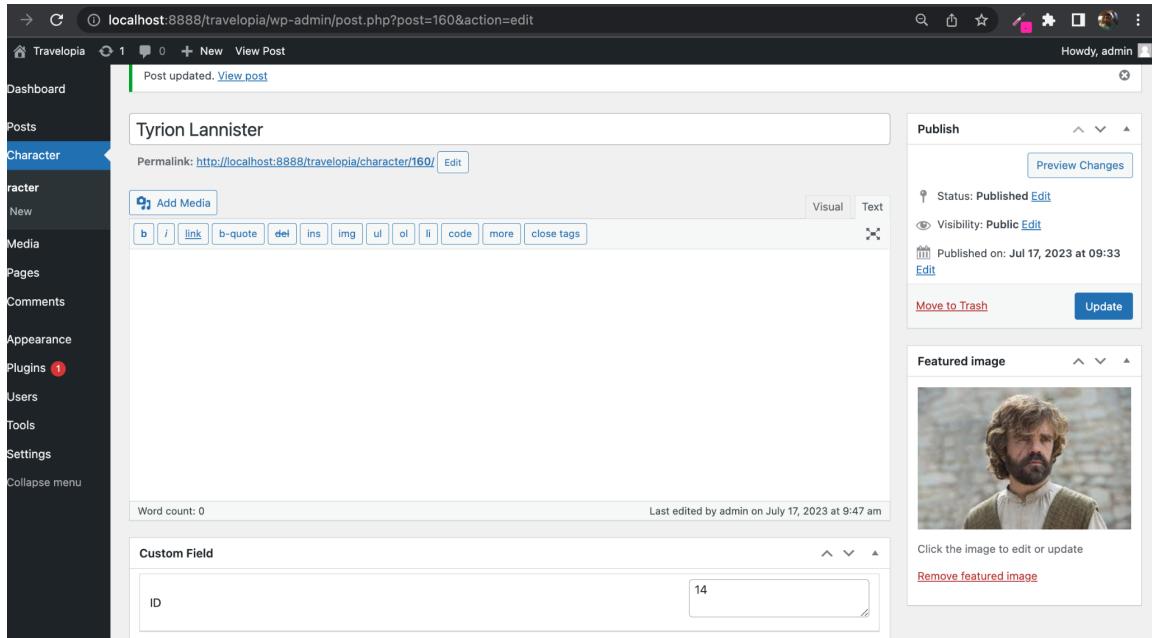
- Dashboard
- Posts
- Character** (selected)
- Character
- Add New
- Media
- Pages
- Comments
- Appearance
- Plugins (1)
- Users
- Tools
- Settings
- Collapse menu

The main content area has a title "Khal Drogo" and a permalink "http://localhost:8888/travelopia/character/160/". It includes a toolbar with "Add Media" and various rich text buttons (b, i, link, b-quote, del, ins, img, ul, ol, li, code, more, close tags). A "Publish" sidebar on the right shows the post status as "Published", visibility as "Public", and a publish date of "Jul 17, 2023 at 09:33". It also includes options to move to trash or update the post.

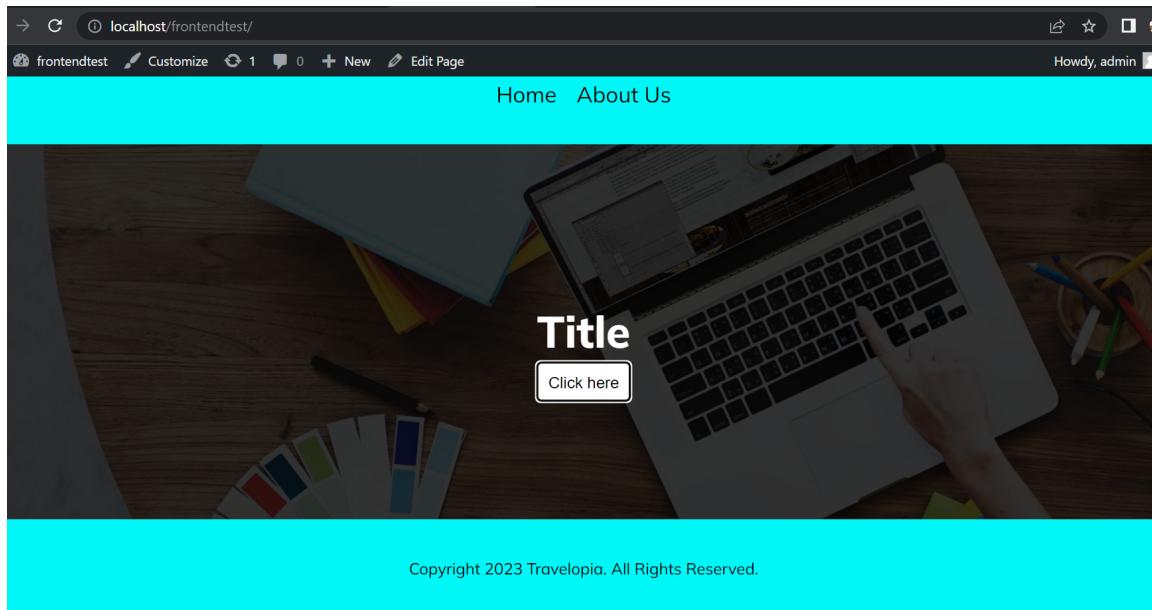
Screenshot 2: Featured Image Selection

This screenshot shows the "Featured image" modal. It lists a single image of Khal Drogo with the file name "khal-drogo.jpg", upload date "July 17, 2023", size "51 KB", dimensions "618 by 412 pixels", and edit and delete options. The modal also includes fields for Alt Text, Title (set to "khal-drogo.jpg"), Caption, and Description, along with a "Set featured image" button.

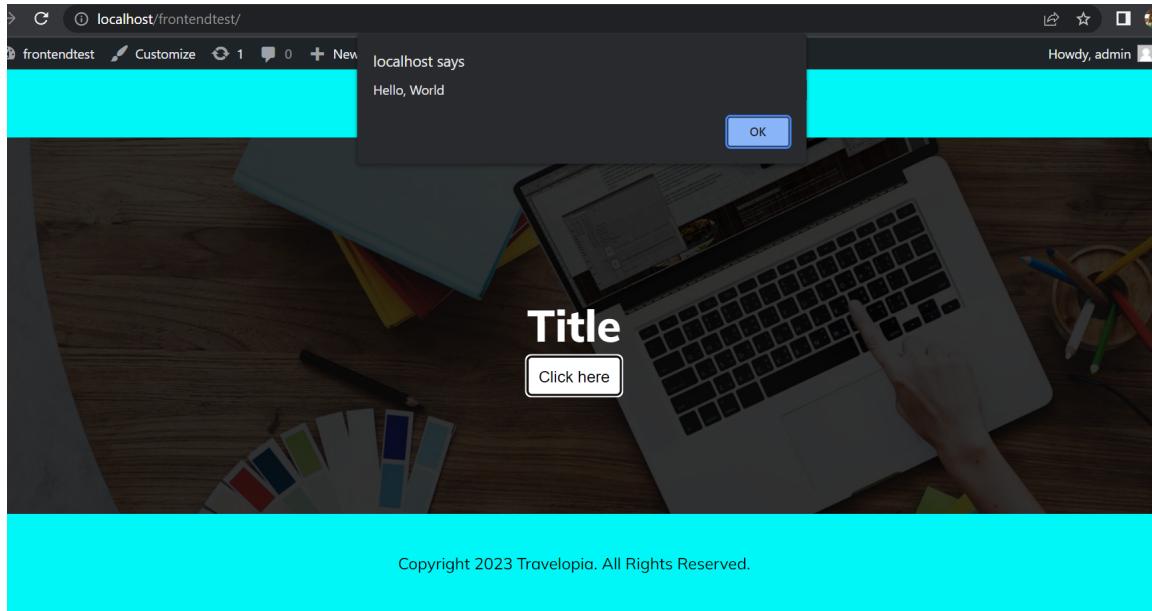
Final Result 



Front End



There is a background image above there is overlay and then title and button



Css used

```
14 * { margin: 0; padding: 0; box-sizing: border-box; }
15 body{ min-height: 100vh; font-family: 'Mulish', sans-serif; }
16 html { min-height: 100%; }
17 img{max-width: 100%}
18 a{ text-decoration:none; color:#000; margin:0 10px }
19 main{ height:calc(100vh - 200px);position: relative; text-align: center; color: #fff; }
20 main img{}
21 main img {
22 position: absolute;
23 width: 100%;
24 height: 100%;
25 left: 0;
26 top: 0;
27 object-fit: cover;
28 }
29 main div []
30 position: absolute;
31 left: 0;display: flex; align-items: center; justify-content: center; height: 100%;
32 top: 0;
33 width: 100%;z-index: 99;background: #rgba(0, 0, 0,0.5);
34 ]
35 h1 {
36 font-size: 44px !important;
37 font-weight: 800;
38 }
39 button {cursor: pointer;
40 background: #fff;
41 position: absolute;
42 color: #000;
43 padding: 10px;
44 font-size: 16px;
45 border-radius: 5px;
46 height: 40px;
47 transform: translateY(50%);
48 margin-top: 60px !important;
49 }header,footer{height: 100px;background: #00F7F7; display: flex !important; align-items: center; justify-content: center; text-align: c
```

```

        color: #000;
        padding: 10px;
        font-size: 16px;
        border-radius: 5px;
        height: 40px;
        transform: translateY(50%);
        margin-top: 60px !important;
    }header,footer{height: 100px;background: #00F7F7; display: flex !important; align-items: center; justify-content: center; text-align: center; .tab,.mobile{display: none;}}
```

```

    @media (max-width: 1300px){

        header a {font-size: 22px}
        img{max-width: 100%}
    }

    /* Media Query */

    @media (max-width: 767px){
        header a {font-size: 20px}
        footer{font-size: 16px}
        .desktop{display: none;}
        .tab{display: block;}
    }

    @media (max-width: 420px){
        header a {font-size: 18px}img{max-width: 100%}
        .desktop,.tab{display: none;}
        .mobile{display: block;}
    }

```

Media query used for responsive screen

Main div css is used for translucent overlay and button css is added for button

Button is added with name Click here

Added alert box through custom js with the help of enqueue script

```

index.php
themes
└ travlopia-theme
    | > img
    | > js
    # custom.css
    # footer.php
    # functions.php
    # header.php
    # index.php
    # page.php
    # screenshot.png
    # single.php
    # style.css
    # index.php
    > upgrade
    > uploads
    # index.php
    > wp-includes

```

```

15     register_nav_menu('header-menu',__( 'Header Menu' ));
16 }
17 // Hook to the init action hook, run our navigation menu function
18 add_action( 'init', 'add_Main_Nav' );
19
20
21 // Enqueue custom js and css for travlopia test
22 function travlopia_test_scripts() {
23     wp_enqueue_style('travlopia-css', get_bloginfo('template_directory').'/custom.css');
24     wp_enqueue_script( 'travlopia-script', get_stylesheet_directory_uri() . '/js/custom.js', array( 'jquery' ) );
25 }
26 add_action( 'wp_enqueue_scripts', 'travlopia_test_scripts' );
27
28
29 /* Author : Nasir Khan
30 * Description : Function to create custom field under post
31 */
32
33
34
35     function character_add_meta_box( $post ){
36         add_meta_box( 'character_meta_box', 'Custom Field', 'character_build_meta_box', 'character', 'normal', 'low' );

```

Custom css added through enqueue script

```
... # custom.css 1 banner.jpg index.php page.php functions.php JS custom.js X
wp-content > themes > MyTheme > js > JS customjs > ...
1  jQuery( document ).ready( function( $ ) {
2      // $() will work as an alias for jQuery() inside of this function
3
4      /* Function to get alert box on click of click here button */
5      $('body').on('click', '.click_button', function() {
6          alert("Hello, World");
7      );
8
9
10
11
12
13
14
15
16
17 } );
```

In 17. Col 5. Spaces: 4. UTF-8. CRLF. {} JavaS

For setting up

<http://localhost/frontendtest>

Frontend - folder name used

Technology : MySql, Wordpress, Php