**E-Commerce application on Cloud Foundry**

**Phase-4 :**

**Back end integrations:**

-> we have used laravel for the back end.

//Logic to add products to cart:

function AddToCart($id) {

$product = Product::find($id);

$user\_id = Auth::user()->id;

Cart::create([

'user\_id' => $user\_id,

'product\_id'=> $product->id,

'product\_name' => $product->name,

'product\_price' => $product->price,

]);

return back();

}

//Logic to checkout

function CheckOut() {

$user\_id = Auth::user()->id;

$carts = Cart::where('user\_id', Auth::user()->id)->select('user\_id', 'product\_id', 'product\_name', 'product\_price')->get();

$array = json\_decode($carts, true);

DB::table('orders')->insert($array);

Cart::where('user\_id', Auth::user()->id)->delete();

return redirect('/home');

}

//Simple logic to display products from database:

@php

$data = \App\Models\Product::all();

@endphp

@foreach($data as $item)

<h6>{{ $item->name }}</h6>

<h5>{{ $item->price }}</h5>

@endforeach

//Logic to authenticate users

class RegisterController extends Controller

{

use RegistersUsers;

@var string

protected $redirectTo = RouteServiceProvider::HOME;

@return void

public function \_\_construct()

{

$this->middleware('guest');

}

@param array $data

@return \Illuminate\Contracts\Validation\Validator

protected function validator(array $data)

{

return Validator::make($data, [

'name' => ['required', 'string', 'max:255'],

'email' => ['required', 'string', 'email', 'max:255', 'unique:users'],

'password' => ['required', 'string', 'min:8', 'confirmed'],

]);

}

/\*\*

\* Create a new user instance after a valid registration.

\*

\* @param array $data

\* @return \App\Models\User

\*/

protected function create(array $data)

{

return User::create([

'name' => $data['name'],

'email' => $data['email'],

'mobile' => $data['mobile'],

'address' => $data['address'],

'password' => Hash::make($data['password']),

]);

}

}

class LoginController extends Controller

{

use AuthenticatesUsers;

@var string

protected $redirectTo = RouteServiceProvider::HOME;

@return void

public function \_\_construct()

{

$this->middleware('guest')->except('logout');

}

}