REAL TIME NOTIFICATION PASOS A SEGUIR:

1. Crear el proyecto:

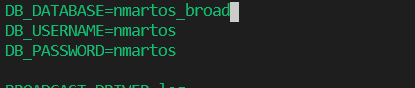
composer create-project laravel/laravel demobroadcast

cd demobroadcast

chmod 777 storage -R

chmod 777 bootstrap -R

modificar fitxer .env



i la url.

php artisan migrate

composer require laravel/breeze - - dev

php artisan breeze:install (este comando te dira que instales unas cosas que son:)

npm install

npm run dev

ESTA CREACIÓN DE UNA VISTA QUE MUESTRE UNA LINEA DE TEXTO EN UNA NUEVA PESTAÑA XXX.BLADE.PHP PUEDO APARECER EN EL EXAMEN

1. Creación de la aplicación

Routes🡪 web.php

Añadir

Route::get('home',[HomeController::class, 'index'])->name("home");

Resources🡪views🡪dashboard.blade.php

 Añadir

<a href="{{route("home")}}"><br />TEST BROAD</a>

Comando para crear el controlador:

php artisan make:controller HomeController -r

este controlador esta en app🡪Http🡪Controllers🡪HomeController.php

Añadir a routes 🡪 web.php

use App\Http\Controllers\HomeController;

Crear una vista en resources🡪views🡪home.blade.php

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Document</title>

</head>

<body>

    Nova vista

</body>

</html>

Y en app🡪Http🡪Controllers🡪HomeController.php he añadido esto para que muestre la vista.

    public function index()

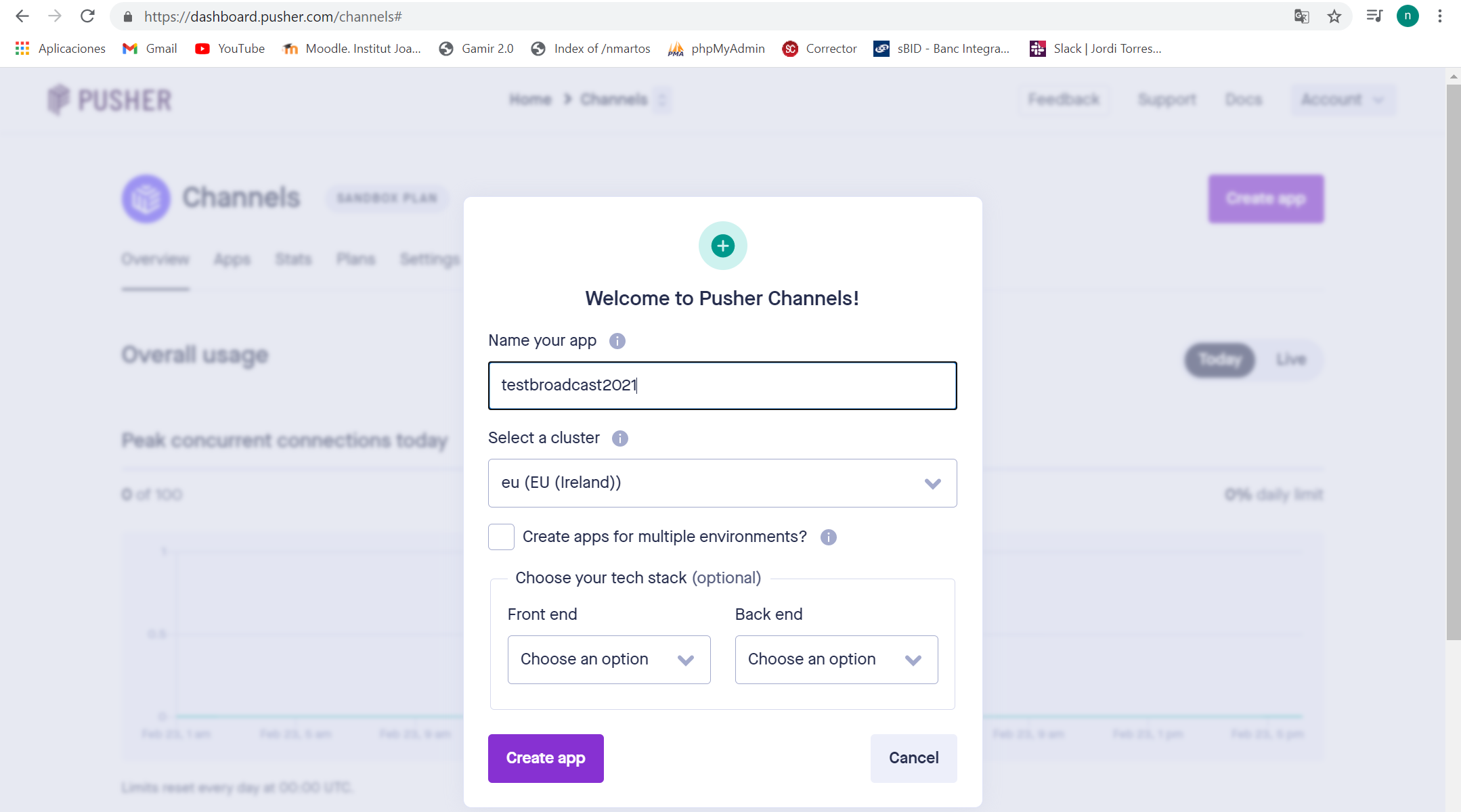
    {

        //

        return view("home");

    }

En PUSHER página web crear el nuevo proyecto:



Dentro de esta web hay un apartado llamado app keys, copiar lo que nos da.

Nos vamos a .env y lo enganchamos

|  |  |
| --- | --- |
| Tenemos esto | Se convierte en esto |
| app\_id = "1161007"  key = "0c58b5efbe489c8636c4"  secret = "f440c2d1b743fb449812"  cluster = "eu" | PUSHER\_APP\_ID=1161007  PUSHER\_APP\_KEY=0c58b5efbe489c8636c4  PUSHER\_APP\_SECRET=f440c2d1b743fb449812  PUSHER\_APP\_CLUSTER=eu |

Activar broadcasting:

Dentro del fitxer .env modificar lo siguiente:

BROADCAST\_DRIVER=pusher

Config🡪app.php

Descomentar línea ~ 174:

App\Providers\BroadcastServiceProvider::class

Instalar pusher:

composer require pusher/pusher-php-server "~4.0"

npm install --save-dev laravel-echo pusher-js

ir a la carpeta resources🡪js🡪bootstrap.js

Descomentar todo este párrafo y añadir la línea de authEndpoint.

import Echo from 'laravel-echo';

window.Pusher = require('pusher-js');

window.Echo = new Echo({

    broadcaster: 'pusher',

    key: process.env.MIX\_PUSHER\_APP\_KEY,

    cluster: process.env.MIX\_PUSHER\_APP\_CLUSTER,

    forceTLS: true,

    authEndpoint: 'https://dawjavi.insjoaquimmir.cat/nmartos/M07/UF2.GENERACI%C3%93%20DIN%C3%80MICA%20DE%20P%C3%80GINES%20WEB%20%20-%20%20UF3.T%C3%88CNIQUES%20D%27ACC%C3%89S%20A%20DADES/demobroadcast/public/broadcasting/auth'

});

Dentro del fichero webpack.mix.js

Añadir línea:

require("dotenv").config();

Esto lo que hace es coger totas las dadas de .env y pasárselas a este fichero

Como se han hecho cambios en un fichero javascript tenemos que compilarlos de manera que:

Npm run dev

Crear una tabla: en este caso llamada messages, con la siguiente comanda:

php artisan make:migration create\_messages\_table

dentro de database🡪migrations🡪 créate\_messages\_table.php

añadir:

    public function up()

    {

        Schema::create('messages', function (Blueprint $table) {

            $table->id();

            $table->string('from');

            $table->string('to');

            $table->string('messages');

            $table->timestamps();

        });

    }

Ejecutar comanda:

Php artisan migrate

Crear el model message y el controlador Message

php artisan make:model Message

php artisan make:controller MessageController -r

Crear un evento (enviar info en un canal):

php artisan make:event NewMessageNotification

app🡪events🡪NewMessageNotification.php

Qeuda de tal forma, subrayado son cosas que he añadido o modificado

<?php

namespace App\Events;

use Illuminate\Broadcasting\Channel;

use Illuminate\Broadcasting\InteractsWithSockets;

use Illuminate\Broadcasting\PresenceChannel;

use Illuminate\Broadcasting\PrivateChannel;

use Illuminate\Contracts\Broadcasting\ShouldBroadcast;

use Illuminate\Foundation\Events\Dispatchable;

use Illuminate\Queue\SerializesModels;

use Illuminate\Contracts\Broadcasting\ShouldBroadcastNow;

use App\Models\Message;

class NewMessageNotification implements ShouldBroadcastNow

{

    use Dispatchable, InteractsWithSockets, SerializesModels;

    public $message = null;

    /\*\*

     \* Create a new event instance.

     \*

     \* @return void

     \*/

    public function \_\_construct(Message $message)

    {

        //

        $this->message = $message;

    }

    /\*\*

     \* Get the channels the event should broadcast on.

     \*

     \* @return \Illuminate\Broadcasting\Channel|array

     \*/

    public function broadcastOn()

    {

        return new PrivateChannel('user.'.$this->message->to);

    }

}

Dentro de routes🡪channel.php

Añadir:

Broadcast::channel('user.{toUserId}', function ($user, $toUserId) {

    return $user->id == $toUserId;

});

app🡪Http🡪Controllers🡪HomeController.php

Añadir lo subrayado:

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

use Illuminate\Support\Facades\Auth;

class HomeController extends Controller

{

    /\*\*

     \* Display a listing of the resource.

     \*

     \* @return \Illuminate\Http\Response

     \*/

    public function index()

    {

        //

        $data["user\_id"] = Auth::user()->id;

        return view("home", $data);

    }

Añadir en resources🡪views🡪home.blade.php

<body>

    Nova vistaaa {{$user\_id}}

</body>

Aquí saldrá el numero del identificador del usuario:

PREGUNTA QUE PUEDE APARECER EN EL EXAMEN:

Que hace este bloque, pregunta examen:

    {

        // message is being sent

        $message = new Message;

        $message->setAttribute('from', 1);

        $message->setAttribute('to', 1);

        $message->setAttribute('message', 'Demo message from user 1 to user 1');

        $message->save();

Crea un insert a la taula de dades, és a dir inserta un missatge amb aquest valors i afegeix una entrada a la DB.

Añadir a app🡪Http🡪Controllers🡪HomeController.php

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

use Illuminate\Support\Facades\Auth;

use App\Events\NewMessageNotification;

use App\Models\Message;

class HomeController extends Controller

{

    /\*\*

     \* Display a listing of the resource.

     \*

     \* @return \Illuminate\Http\Response

     \*/

    public function index()

    {

        //

        $data["user\_id"] = Auth::user()->id;

        return view("home", $data);

    }

    public function send(){

        // message is being sent

        $message = new Message;

        $message->setAttribute('from', 1);

        $message->setAttribute('to', 1);

        $message->setAttribute('messages', 'Demo message from user 1 to user 1');

        $message->save();

        // want to broadcast NewMessageNotification event

        event(new NewMessageNotification($message));

   return "sms enviado...";

    }

Añadir una ruta a web.php

Route::get('message/send',[HomeController::class, 'send']);

Añadir a resources🡪views🡪home.blade.php

    <!-- Scripts -->

    <script src="{{ asset('js/app.js') }}" defer></script>

    <script id="functions" user-id="{{ $user\_id }}" src="{{ asset('js/functions.js') }}" defer></script>

Crear el fichero siguiente dentro de public/js crear **touch functions.js** el cual contiene:

window.onload = function() { init(); };

function init(){

    var script\_tag = document.getElementById('functions')

    var user\_id = script\_tag.getAttribute("user-id");

    Echo.private('user.'+user\_id)

    .listen('NewMessageNotification', (e) => {

    alert(e.message.messages);

    });

}

// la funcio init es crida quan s'ha carregat la pagina, és a dir, la vista.

Para probar en pusher que todo está ok

/\*

App\Events\NewMessageNotification

private-user.1

{

"messages": {

"from": 1,

"to": 1,

"message": "Demo message from user 1 to user 2",

"updated\_at": "2021-02-22T08:15:32.000000Z",

"created\_at": "2021-02-22T08:15:32.000000Z",

"id": 81

}

}

\*/

Pagina web para enviar los mensajes:

<http://dawjavi.insjoaquimmir.cat/nmartos/M07/UF2.GENERACI%C3%93%20DIN%C3%80MICA%20DE%20P%C3%80GINES%20WEB%20%20-%20%20UF3.T%C3%88CNIQUES%20D%27ACC%C3%89S%20A%20DADES/demobroadcast/public/message/send>