

Examen Projectes Web

Juny 2019

Prerequisites

Per poder implementar aquest exercici necessitaràs un entorn de desenvolupament que contingui:

1. Un servidor web (Nginx o Apache)
2. PHP7
3. MySQL
4. Composer

Introducció

Com cada any, la gent del Social Club de la Salle té molts problemes per programar els horaris de les activitats del Sallefest, que l'any que ve serà del **20 al 23 d'Abril**, ens han demanat que els hi fem un calendari per portar un control de les activitats. Totes les seccions de l'aplicació es descriuen en detall a continuació.

Requeriments

Pàgina de benvinguda (1 punts)

Ruta	Mètode
/	GET

Aquesta pàgina ja està implementada en el template que podeu descarregar de l'Estudy. Bàsicament conté un llistat amb totes les activitats que han sigut registrades a l'aplicació. Per cada una de les activitats, es mostra el nom, el preu, la data de registre, la data de la activitat i la seva descripció.

Als usuaris que tenen la sessió iniciada, se'ls mostra un botó que els permet esborrar les seves activitats. Als usuaris que no han iniciat cap sessió, se'ls mostra dos nous enllaços en el menú de navegació:

1. **Registre:** ha de redirigir els usuaris al formulari de registre `/register`.
2. **Login:** ha de redirigir els usuaris al formulari de login `login`.

Un cop l'usuari ha iniciat la sessió, cal amagar ambdós enllaços.

Si obriu l'arxiu `home.html.twig`, veureu que el template espera rebre dues variables:

1. `logged`: és un booleà que indica si l'usuari té la sessió iniciada o no
2. `activities`: un llistat amb totes les activitats que s'han registrat a l'aplicació

Registre (2 punts)

Ruta	Mètode
<code>/register</code>	GET
<code>/register</code>	POST

Quan els usuaris accedeixen a la URL `/register` han de veure un formulari amb els camps següents:

1. **username**

- El camp no pot estar buit
- Ha de contenir una màxim de 10 caràcters
- No pot contenir cap número

2. **email**

- Ha de tenir un format correcte

3. **password**

- Ha de contenir entre 6 i 12 caràcters

Un cop l'usuari faci click al botó de *submit* el formulari s'haurà d'enviar a `/register` fent servir el mètode **POST**. Tots els camps hauran de ser validats al Controlador abans de registrar l'usuari.

Si hi ha cap error, s'haurà de mostrar novament el formulari marcant en cada camp els errors que s'han trobat durant la validació.

Nota: No heu de d'implementar cap validació amb JavaScript.

Per crear la taula d'usuaris, heu de fer servir el següent codi SQL:

```
CREATE TABLE `user` (  
  `id` int(10) unsigned NOT NULL AUTO_INCREMENT,  
  `username` varchar(10) NOT NULL,  
  `email` varchar(100) NOT NULL,  
  `password` varchar(12) NOT NULL,  
  `created_at` datetime DEFAULT NULL,  
  `updated_at` datetime DEFAULT NULL,  
  PRIMARY KEY (`id`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

Login (1 punts)

Ruta	Mètode
/login	GET
/login	POST

Quan un usuari accedeixi a la pàgina `/login`, haurà de veure un formulari amb els següents camps:

1. username/email Els usuaris han de poder fer servir tant el nom d'usuari com el correu per tal d'iniciar la seva sessió.
2. password

Un cop l'usuari faci click en el butó de *submit* el formulari s'haurà d'enviar a `/login` fent servir un mètode **POST**. Tots el camps han de ser validats en el Controlador abans d'iniciar la sessió de l'usuari. Si hi ha cap error, s'haurà de mostrar novament el formulari de login tot indicant els errors que s'han produït.

Registre d'activitats (3 punts)

Ruta	Mètode
/activities	GET
/activities	POST

Quan els usuaris accedeixin a la pàgina `/activities`, heu de mostrar un formulari amb els següents camps:

1. **name**
 - Ha de contenir un màxim de 20 caràcters
2. **description**

- Ha de ser un *input* de tipus *text area*

3. price

- Ha de ser un *enter*

4. schedule

- Ha de ser un *input* de tipus *date*

Tots els usuaris que intentin accedir a aquesta secció sense haver iniciat cap sessió, hauran de ser redirigits cap a la pàgina de `login`.

Un cop l'usuari faci click en el botó de *submit* el formulari s'haurà d'enviar a `/activies` fent servir un mètode **POST**. Tos els camps s'hauran de validar en el Controlador abans de registrar la tasca a la base de dades. Si hi ha cap error, s'haurà de mostrar novament el formulari de l'activitat amb tots els errors.

S'ha de tenir en compte que **no podràn haver-hi dues activitats a la mateixa hora** i s'ha de comprovar que l'activitat estigui dins de la setmana del **20 al 23 d'Abril del 2020**.

Per crear la taula **activies**, heu de fer servir el següent *snippet* de SQL:

```
CREATE TABLE `activity` (  
  `id` int(10) unsigned NOT NULL AUTO_INCREMENT,  
  `id_user` int(10) unsigned NOT NULL,  
  `name` varchar(20) NOT NULL,  
  `description` text NOT NULL,  
  `price` int NOT NULL,  
  `created_at` datetime DEFAULT NULL,  
  `activity_time` datetime DEFAULT NULL,  
  PRIMARY KEY (`id`),  
  KEY `user_task` (`id_user`),  
  CONSTRAINT `user_task` FOREIGN KEY (`id_user`) REFERENCES `user` (`id`) ON DELET  
E CASCADE ON UPDATE CASCADE  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

Podeu tenir certes entrades a la taula per fer proves, podeu fer servir el codi següent:

```
INSERT INTO user VALUES (1, 'root', 'admin@gmail.com', 'root', DATE('2020-4-20 01:00:00'), DATE('2020-4-20 01:00:00'));
INSERT INTO activity VALUES (1, 1, 'Futbol', 'Tirar pilotetes a la porteria', 3, DATE('2020-4-20 01:00:00'), DATE('2020-4-20 11:00:00'));
INSERT INTO activity VALUES (2, 1, 'Volley', 'Tirar pilotetes per sobre una xarxa', 2, DATE('2020-4-20 01:00:00'), DATE('2020-4-22 11:00:00'));
INSERT INTO activity VALUES (3, 1, 'Basquet', 'Tirar pilotetes al aro', 5, DATE('2020-4-20 01:00:00'), DATE('2020-4-22 11:00:00'));
INSERT INTO activity VALUES (4, 1, 'Cupflip', 'Girar gotets', 2, DATE('2020-4-20 01:00:00'), DATE('2020-4-21 11:00:00'));
INSERT INTO activity VALUES (5, 1, 'Beerpong', 'Tirar pilotetes a la birra', 5, DATE('2020-4-20 01:00:00'), DATE('2020-4-23 11:00:00'));
```

Pàgina de visualització d'una activitat (2 punt)

Route	Method
/activies/{id}	GET

Si l'usuari que intenta accedir a aquesta secció no ha iniciat cap sessió, s'haurà de mostrar tots els atributs de l'activitat (exceptuant l'id).

Si l'usuari té una sessió iniciada, s'haurà de mostrar tots els atributs de l'activitat (exceptuant l'id) **i afegir un botó que permeti eliminar la activitat**.

Eliminar una activat (1 punt)

Ruta	Mètode
/activies/{id}/delete	POST

Si l'usuari intenta eliminar una activitat que no és seva s'haurà de mostrar un missatge d'error avisant l'usuari.

Si la activitat ha estat eliminada correctament s'ha de redirigir a l'usuari cap al calendari d'activitats.

Lliurament de l'entrega

- Disposeu de dues hores per implementar l'exercici
- Podeu consultar i fer servir qualsevol recurs
- No es pot modificar l'estructura de les taules que es faciliten en l'enunciat
- Cal fer servir Slim per implementar l'exercici
- Al finalitzar, cal penjar un **.zip** que contingui el codi, esborrant la carpeta *vendor*, a l'Estudy*

- Qualsevol comentari que creieu que és important de cara a la correcció del vostre examen, el podeu afegir en un document README en el mateix ZIP de l'entrega