

Zenit

Ett CMS för elever och lärare på YH utbildningar

Namn: Tillmann Weimer

Klass: WCMS 18

Handledare: Marcus Ekelund

Datum: 07.05.2020

Innehållsförteckning

- Syfte och mål
- Avgränsningen
- Arbetets förlopp
- Efter tömning av databasen (*php artisan migrate:fresh*)
- Avgränsning (Perspektivering / Vidareutveckling)
- Erfarenheter
- Referenser och källförteckning

Syfte och mål

Mitt syfte är att skapa ett system som på sikt skulle kunna ersätta det nuvarande Zenit (*Ping Pong*). Inte alla använder sig av det vilket har lett till en del missuppfattningar vilka i sin tur har påverkat informationsflödet mellan elever och lärare. Dessutom har det nuvarande Zenit rätt så stora brister när det kommer till användarvänlighet.

Ramverk

Jag valde en objektorienterad tillgång med Laravel 6 som MVC ramverk. Laravel ger goda lösningar när det gäller autentisering och säkerhet.

Uppbyggnad

Min applikation bygger på autentisering med två olika typer user. De har olika befogenheter.

1. Admin

I det följande vill jag göra redo för vilka funktionaliteter en användare med Admin rättigheter har. Denna person skulle kunna vara en lärare på ett utbildningsinstitut.

- En användare med Admin behörighet kan ladda upp eller ändra en site logo som visas på vänster sida i navbaren. Den kan dessutom skapa programs (*utbildningar*) och redigera courses och news. En Admin användare kan se inkommande projekt på när den klickar på 'Admin' länken i navbaren, då den kan skriva en kommentar och via en checkbox kan avgöra om projektet är komplett (*completed*).

2. Member

Användaren med member rättigheter skulle kunna vara en elev på en utbildning.

- Den kan antingen ladda upp ett projekt via en länk under (*courses*) i Akkordion på vänster sida eller när den klickar på själva projektet för att få visat det på helskärm.
- När en member användare ladda upp ett projekt kan den även skicka länken till en webbsida till sitt projekt om den har en sån
- Om en Admin användare har kommenterat ett projekt kan Member se det när den klickar på (*your page*). Om projektet behöver kompletteras visas en röd button under det med texten (*complete your project*). Kommentaren visas i det fallet på höger sida med röd bakgrund. Om projektet inte är komplett.

Om Admin har klickat på *completed* visas ingen *complete project* button på Member sidan och bakgrunden är grön.

- Som Member kan man även se en lista över avslutade kurser när man klickar på dropdown 'Privat' och väljer 'Your success'.
- En användare med Member behörigheter kan skriva sitt Motto som visas på sidan efter inloggningen.

Alla inloggade användare kan ladda upp en profilbild som visas både i navbaren och på sidan efter inloggningen.

Arbetets förlopp

1. Jag började mitt projekt med att rita upp databasen vilket dannede udgangspunkt for hvilke model jag skulle behöva. Jag skapade models med kommandon `'php artisan make:model "Project" -r -m -c'`.

'-m' betyder att den automatisk även skapar en migration som i sin tur skapar och utvecklar databasen.

'-r' betyder att man kan skapa en resource controller vilket både underlättar när man ska definera routes. Man behöver inte upprepa sig varje gång man definerar en ny route inom samma Controller.

Samtidigt behöver man då inte själv definera samtliga metoder i sin *Controller*

Jag valde ändå endast att ha två resource controller (*Course och News*) och uppnådde ändå full funktionalitet. Jag gjorde det medvetet med tanken om att skulle vilja lära mig hur routes fungerar.

Detta skulle självklart kunna ändras för att minska storleken av *web.php* filen och inte minst underlätta för potentiella nya utvecklare i fall de skulle ta över och vidareutveckla applikationen.

2. Laravel utmärker sig i att bjudar på goda verktyg när det gäller databashanteringen om man jobbar agilt. Detta ger goda möjligheter i att synka en databas. Jag provade vid flera tillfällen att ändra en kolumn via en migration istället för att ändra direkt i databasen för att träna mig med detta.

Efter tömning av databasen (*php artisan migrate:fresh*)

När man tömmer databasen och registrerar den första användare borde den få admin rättigheter eftersom det behöver skapas ett *program*. Det är endast admin som kan skapa *program*. Ett program ska finnas för att kunna skapa *courses*

Avgränsning (Perspektivering / Vidareutveckling)

1. Man skulle kunna lägga till funktionaliteten att *admin* kan se varje gång när ett projekt har kompletterats i samband med en annan färg. Men med tanken på tidsrammen för detta projekt valde jag att inte lägga till denna funktionalitet

2. Man skulle även kunna vidareutveckla '*your success*' i att lägga till en progressbar eller någon annan animerat lösning som visar hur många procent en elev har avklarat som det finns på det nuvarande Zenit, men med tanken på tidsrammen valde jag att inte implementera denna funktionalitet.
3. Jag har lagt översättningstaggar runt om varje textsträng, således at man senare skulle kunna implementera stöd för översättningen. Men med tanken på tidsrammen för detta projekt valde jag att inte utveckla denna funktionalitet fullt ut.
4. Med tanken på tidsestimeringen för mitt examensarbete som helhet valde jag att användaren med *Member* endast kan ladda upp sina projekt som textmeddelanden. Än så länge finns det ingen filuppladning till detta.
5. När man skapar den första användare läggs det inte till ett *program_id* eftersom det inte finns *program* i databasen. Då måste man lägga till det första *program_id* manuellt i databasen till den första user (*column program_id*).

Med tanken på uppgiftens omfang valde jag inte att lägga till denna funktionalitet. Man skulle kunna lösa detta i att skapa ytteligare en *user_type* med namn *Superadmin*.

Superadmin:

Denna *user_type* skulle exempelvis kunna vara YH styrelsen eller Högskoleverket med rättigheter att skapa *program* och även uppladningen av deras *site_logo*.

Referenser och källförteckning

- <https://laravel.com/>
- <https://laracasts.com/>
- <https://stackoverflow.com/>