COMP 3211 Software Engineering

Software Design Specification

24 November, 2017

<E-learning Course Application Website>

Project members

Tsui Kin Min (15076132D)

Lam Tsun Fung (15067703D)

Ng Tsz Kin (15066736D)

Contents

Part II: Software Design Specification

1.0 Introduction

eBoard is a web-based course registration platform aiming to provide useful courses for teenagers and adults on learning purpose. Users can reduce the time spending on searching desired courses through eBoard. The potential users could be students and adults for academic or advanced studies. eBoard aims to provide non-traditional curricular schooling which users can register their favorite courses for learning with available session.

2.0 Data design

2.1 Internal software data structure

User

ID (P.K.) - The id of the user

Name - The name of the user

Email - The email of the user

Password - The password of the use account

Remember_token - A unique token to decide if the user is remembered by the platform or not

Stripe_id - The id for Stripe (Third party payment platform)

Card_brand - The brand of card used by user

Card_last_four - The last four digit of the card used by the user

Trial_ends_at - The end time of the trial specified by the subscription plan

Created_at - The time of the user joined the platform

Updated_at - The time of the user made modification to his/her account

User Detail

ID (P.K.) - The id of the user's detail

User_id - The id of the user corresponding to that user's detail

legit - Whether or not the user is being banned or not

User Profile

ID(P.K.) - The id of the user's profile

User_id - The id of the user corresponding to that user's profile

Nickname - The nickname of the user used for in-platform representation

Age - The age of the user

Gender - The gender of the user

Mobile - The mobile number of the user

DOB - The date of birth of the user

Country - The nationality of the user

Icon - The URL of the icon specified by the user

Event

ID(P.K.) - The id of the event

User_id - The id of the host as a user

Speaker - The name of the speaker(s)

Location - The location of the event

Venue - The venue of the event

Title - The title of the event

Content - The description of the event

Created_at - The time of the event being created

Updated_at - The time of the event being modified

Admission

ID(P.K.) - The id of the admission

Course_id - The id of the course

User_id - The id of the user

Created_at - The time of the admission being created

Updated_at - The time of the admission being updated

Course

ID(P.K.) - The id of the course

Name - The name of the course

Location - The location of the course

Level - The difficulty level of the course

Duration - The duration of the course

Created_at - The time of the course being created

Updated_at - The time of the course being modifited

Course Detail

ID(P.K.) - The id of course' detail corresponding to that course

Course_id - The id of the course

Discipline_id - The id of the discipline that the course belongs to

Discipline

ID(P.K.) - The id the discipline

Name - The name of the discipline

Created_at - The time of the discipline being created

Updated_at - The time of the discipline being modified

Role

ID(P.K.) - The id of the role

Guard_name - The name of the interface used by the corresponding role so as to act as a filter

Created_at - The time of the role being created

Updated_at - The time of the event being modified

Permission

ID(P.K.) - The id of the permission

Guard_name - The name of the interface used by the corresponding permission so as to act as a filter

Created_at - The time of the permission being created

Updated_at - The time of the permission being modified

Post

ID(P.K.) - The id of the post

Title - The title of the post

Body - The content of the post

Created_at - The time of the post being created

Updated_at - The time of the post being modified

Subscription

ID(P.K.) - The id of the subscription

Stripe_id - The id for Stripe (Third party payment platform)

Stripe_plan - The name of the plan specified by Stripe

Quantity - The quantity of payment specified by user

Trial_ends_at - The end time of the trial specified by the subscription plan

Ends_at - The end time of the whole subscription specified by the subscription plan

Created_at - The time of the subscription being created

Updated_at - The time of the subscription being modified

Notification

ID(P.K.) - The id of the notification

Type - The type of the notification

Notifiable_id - The id of the interface being notified

Notifiable_type - The name of the interface being notified

Data - The data of the notification

Read_at - The time of the notification being read by user

Created_at - The time of the notification being created

Updated_at - The time of the notification being modified

Message

ID(P.K.) - The id of the message

Name - The name of the sender

Email - The email of the sender

Phone - The phone number of the sender

Message - The content of the message

Created_at - The time of the post being created

Updated_at - The time of the post being created

Comment

ID(P.K.) - The id of the comment

Event_id - The id of the event that the comment belongs to

User_id - The id of the user who issue the comment

Comment - The content of the comment

Created_at - The time of the comment being created

Updated_at - The time of the comment being modified

Blacklist

ID(P.K.) - The id of the blacklist

User_id - The id of the user

Ip_address - The ip address of the device used by the suer

Created_at - The time of the blacklist being created

Updated_at - The time of the blacklist being modified

2.2 Global data structure

Session

ID(P.K.) - The id of the session

User_id - The id of the user who granted the corresponding session

Ip_address - The ip address of the device used by the user

User_agent - The agent/browser used by the user

Last_activity - The time of the user's last appearance on the platform

Cookie

Name - The name of the cookie

Content - The content of the cookie

Host - The host/user of the cookie

Path - The path of the cookie

Send_for - The connection which will use the cookie

Expired - The expire time of the cookie

2.3 Temporary data structure

Security Token

CSRF_token - A unique id to prevent Cross-Site Request Forgery

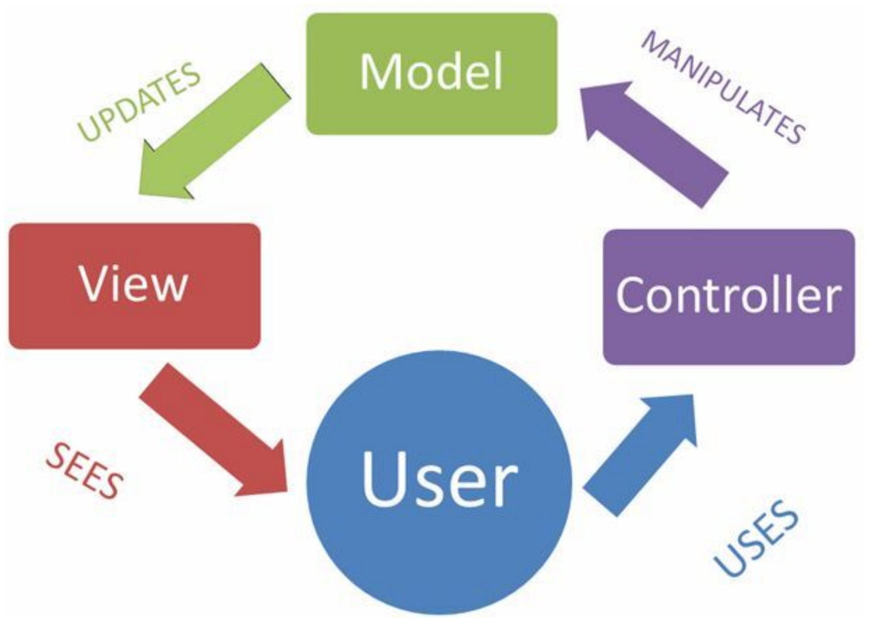
X-CSRF_token - A unique id to generate the CSRF_token

2.4 Database description

The database we are using is MySQL database. It will be created using MySQL 5.7.19. It will hold all of the information entered by the user during the design of the game. This information will be stored in the tables that are listed above in the section titled Internal Software Data Structures.

3.0 Architectural and component-level design

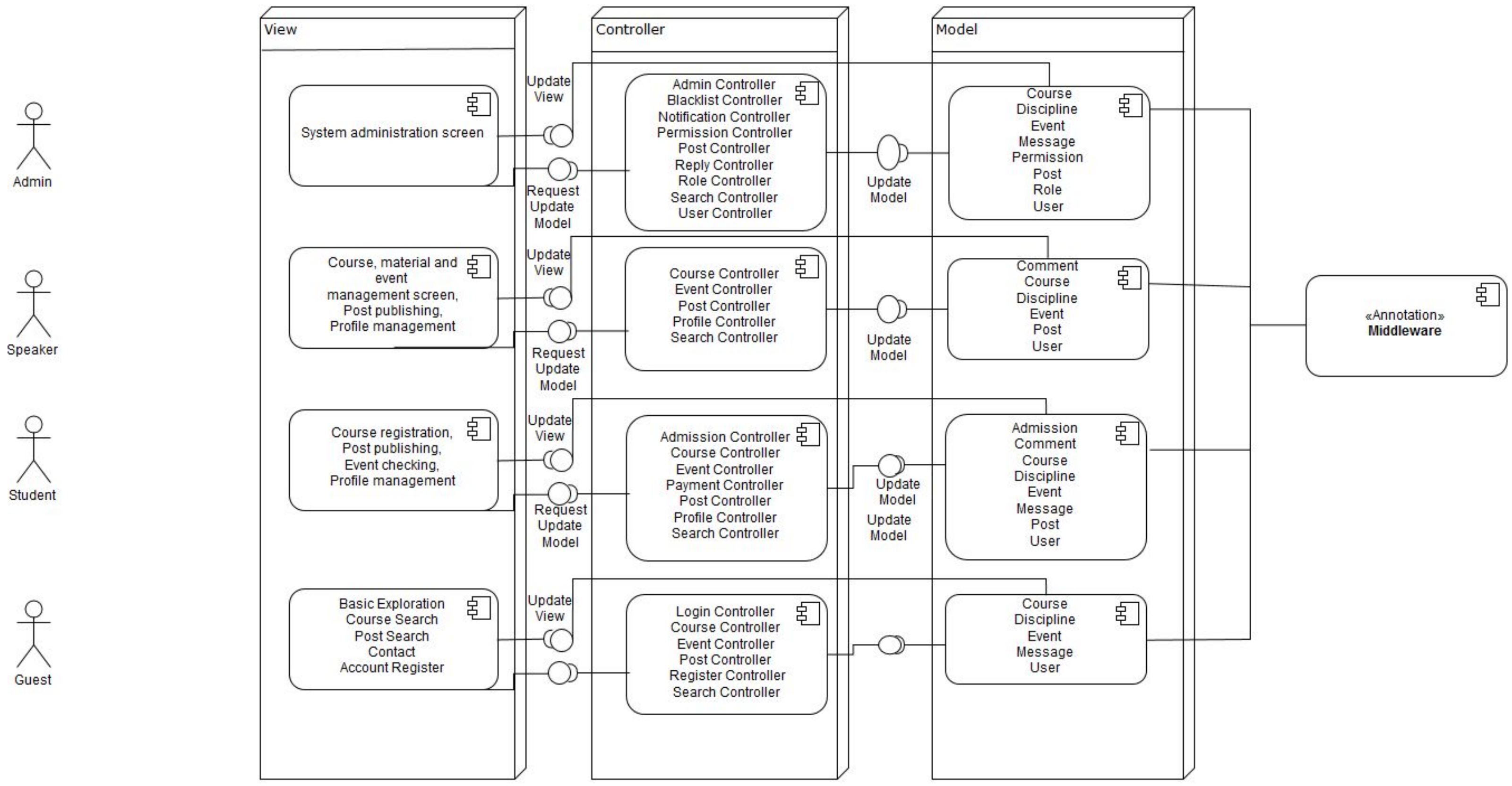
3.1 System Structure



Laravel framework used Model–view–controller (MVC) architecture pattern.
Model abstract database as data, it is associated with database and handle relationships.

View is the content served to the user (displayed in browser) base on changes in model. Controller handle request passing from view with logic then manipulate with model. It may then update view with data retrieved from model.

3.1.1 Architecture diagram


View
Controller
Model
Admin
Speaker
Student
Guest
System administration screen
Course, material and event management screen, Post publishing, Profile management
Course registration, Post publishing, Event checking, Profile management
Basic Exploration Course Search Post Search Contact Account Register
Update View
Request Update Model
Update Model
Admin Controller Blacklist Controller Notification Controller Permission Controller Post Controller Reply Controller Role Controller Search Controller User Controller
Course Controller Event Controller Post Controller Profile Controller Search Controller
Admission Controller Course Controller Event Controller Payment Controller Post Controller Profile Controller Search Controller
Login Controller Course Controller Event Controller Post Controller Register Controller Search Controller
Course Discipline Event Message Permission Post Role User
Comment Course Discipline Event Post User
Admission Comment Course Discipline Event Message Post User
Course Discipline Event Message User
«Annotation» Middleware

3.2 Description for MVC

Laravel components are under Illuminate folder, for example database, view, auth components. Here we treat model, view, controller and database as four main components in our system. There are different MVC components for different use cases for example UserProfile.php (model), UserProfile.blade.php (view) ProfileController.php (controller) for viewing user's profile. There are actually many other components for example middleware which is guarding unauthorized users from accessing routes that are without permissions. However they are ignored here for simplicity.

3.3 Dynamic Behaviour for each use case base on Component MVC

<u>AccountRegistration</u>

Model: User - for inserting new user

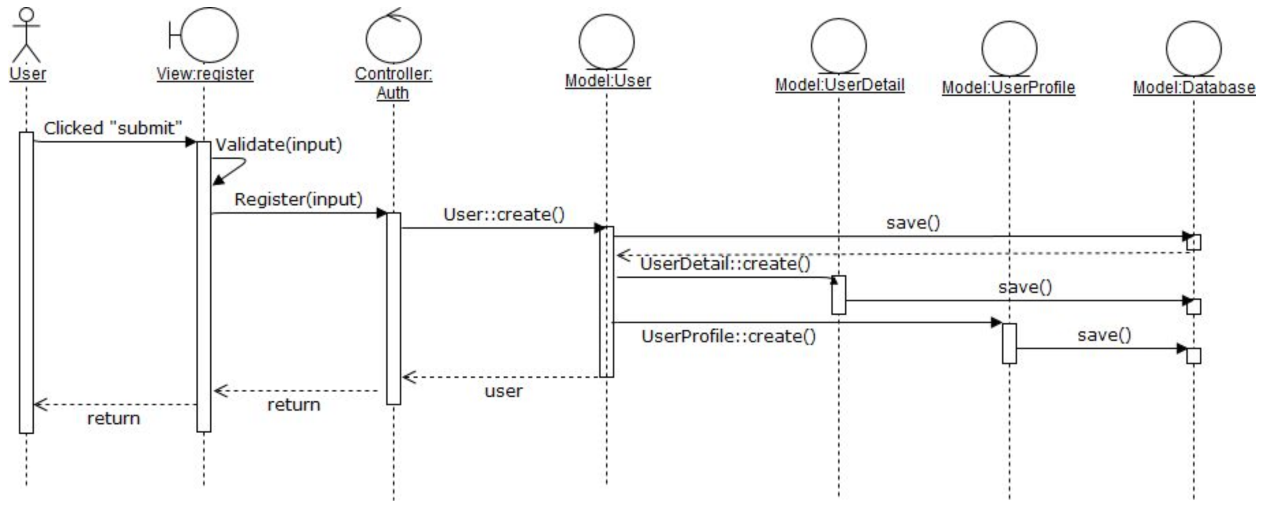
UserDetail - for inserting user status (i.e. set default legit flag as 1)

UserProfile - for inserting basic user info (i.e. gender, age…)

View: register - page that let user register account

Controller: Auth - controller which validate input. It is the default laravel registration component.

(AccountRegistration)



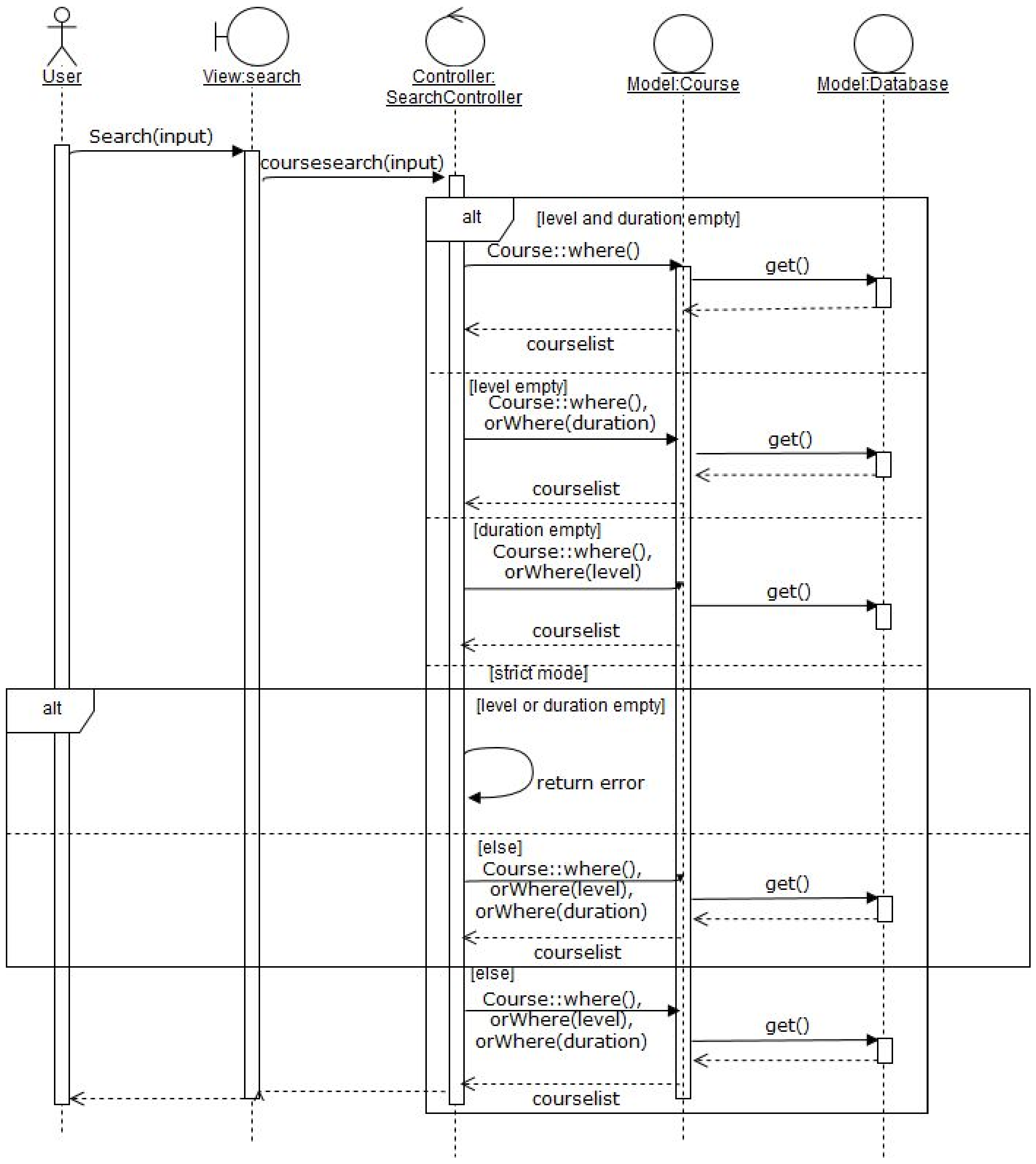
<u>Search</u>

Model: Course - for retrieving information from course table

View: search - page that let user to search

Controller: SearchController- controller which search by conditions. There is a strict mode that if user leave a search field blank it will return error.

(Search)



<u>Payment</u>

Model: User - for retrieving user's information

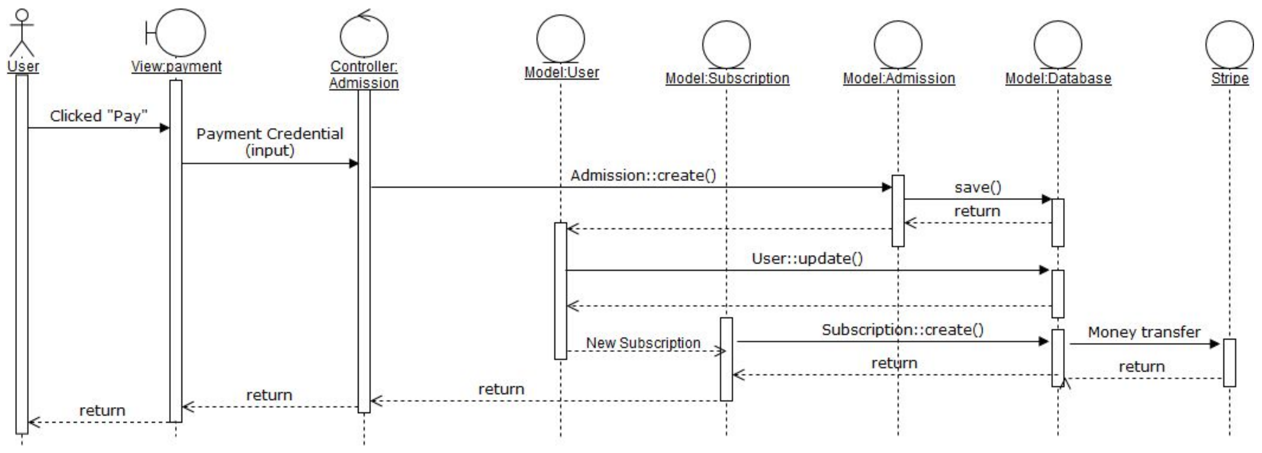
Subscription - for creating new subscription record

Admission - for creating new admission record

View: payment - page that let user to pay for course

Controller: AdmissionController - controller which create new records in databases tables. It will transfer money using stripe api.

(Payment)



UserProfile

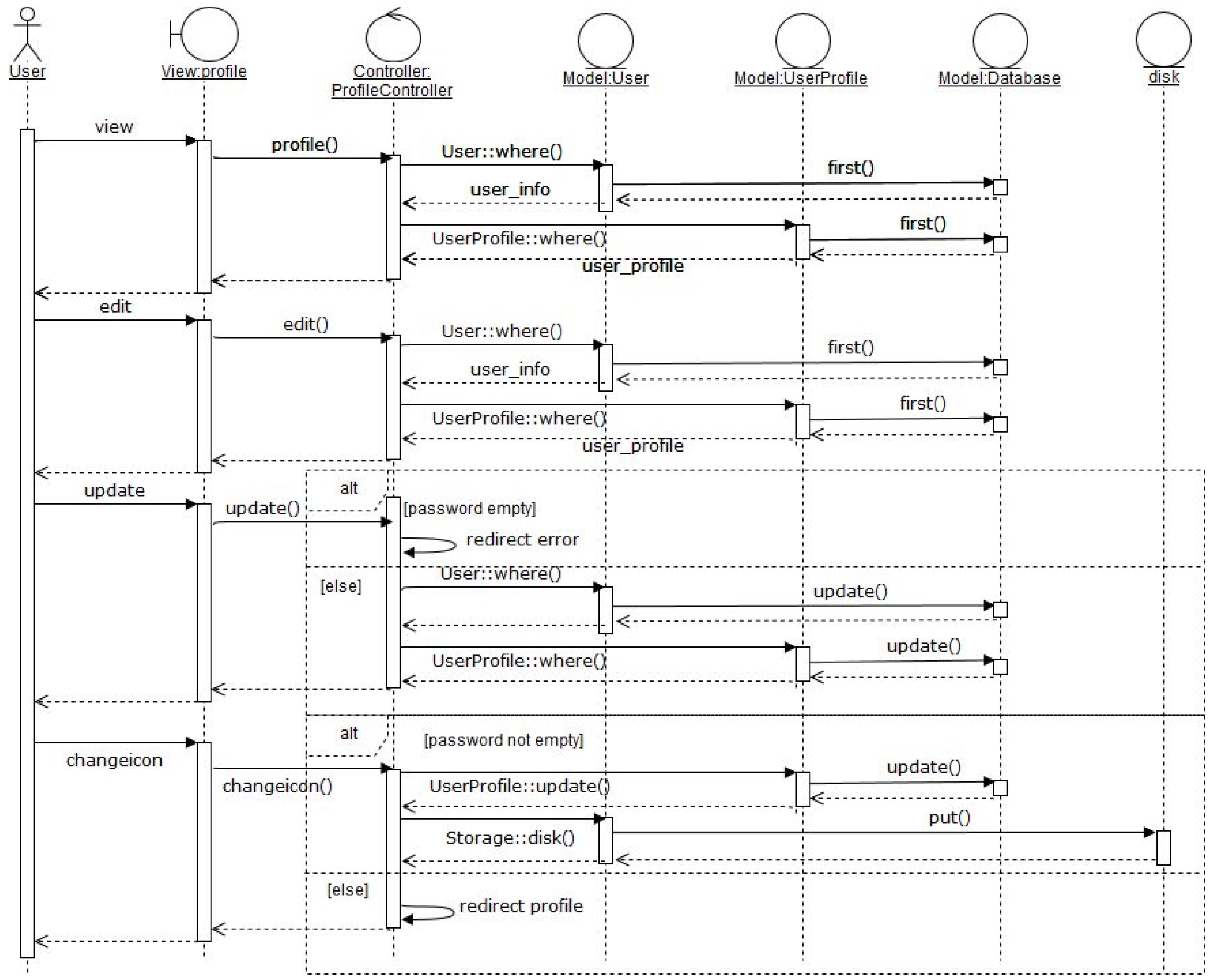
Model: User - for retrieving user's information

UserProfile - for retrieving or updating user's profile

View: profile - page that let user to view profile

Controller: ProfileController - controller which control view, edit, update and changeicon function in profile model.

(User Profile)



<u>Course</u>

Model: Course - for CRUD(create, read, update, delete) operations in course table

Discipline - each course must have one discipline

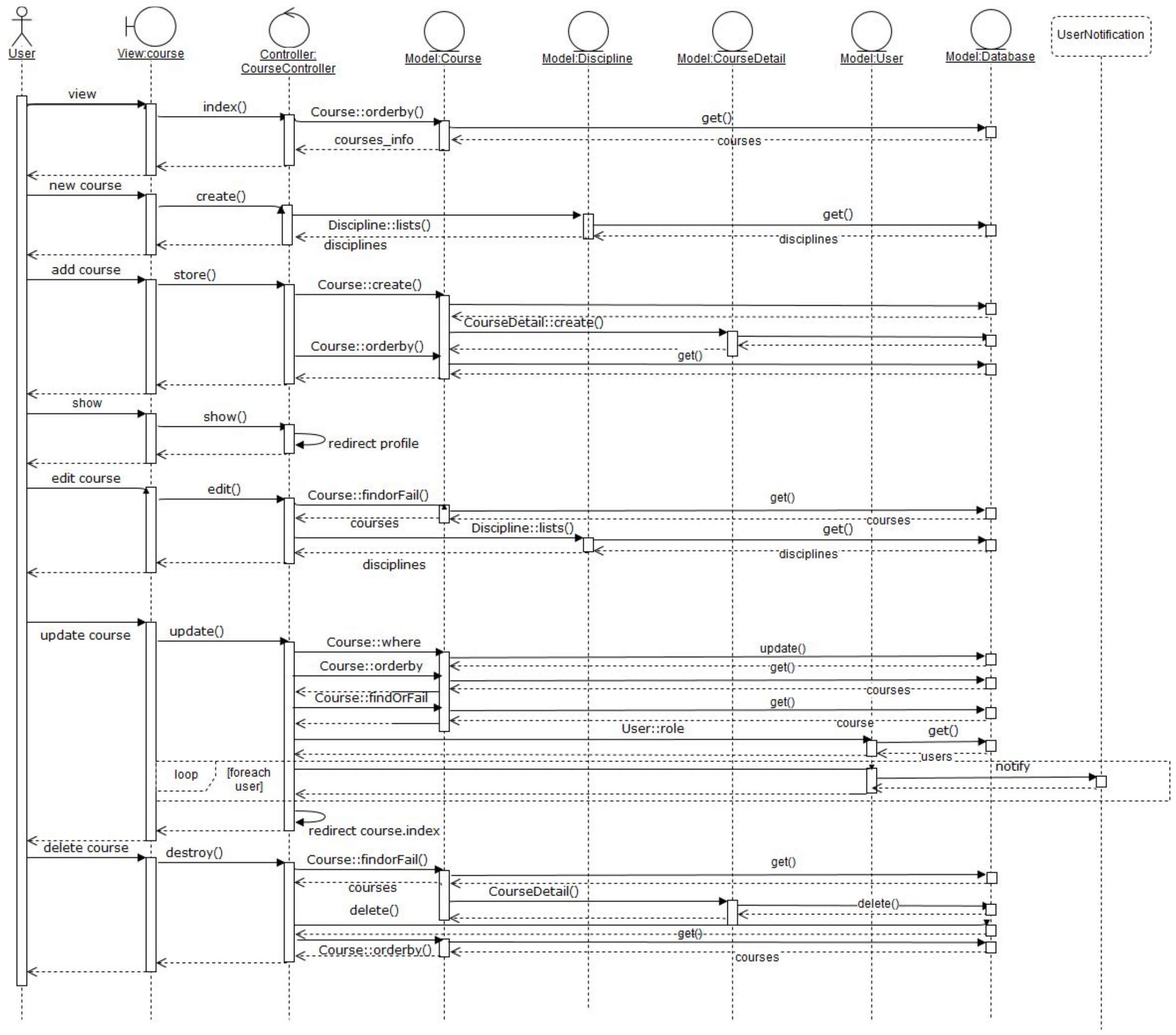
CourseDetail - get all course related informations

User - for retrieving user's information

View: course - we used RESTful resource controllers so there are course.index, course.create, course.store, course.show, course.edit, course.update, course.destroy routes for doing CRUD operations.

Controller: CourseController - controller which implements the seven functions respectively in view.

(Course)



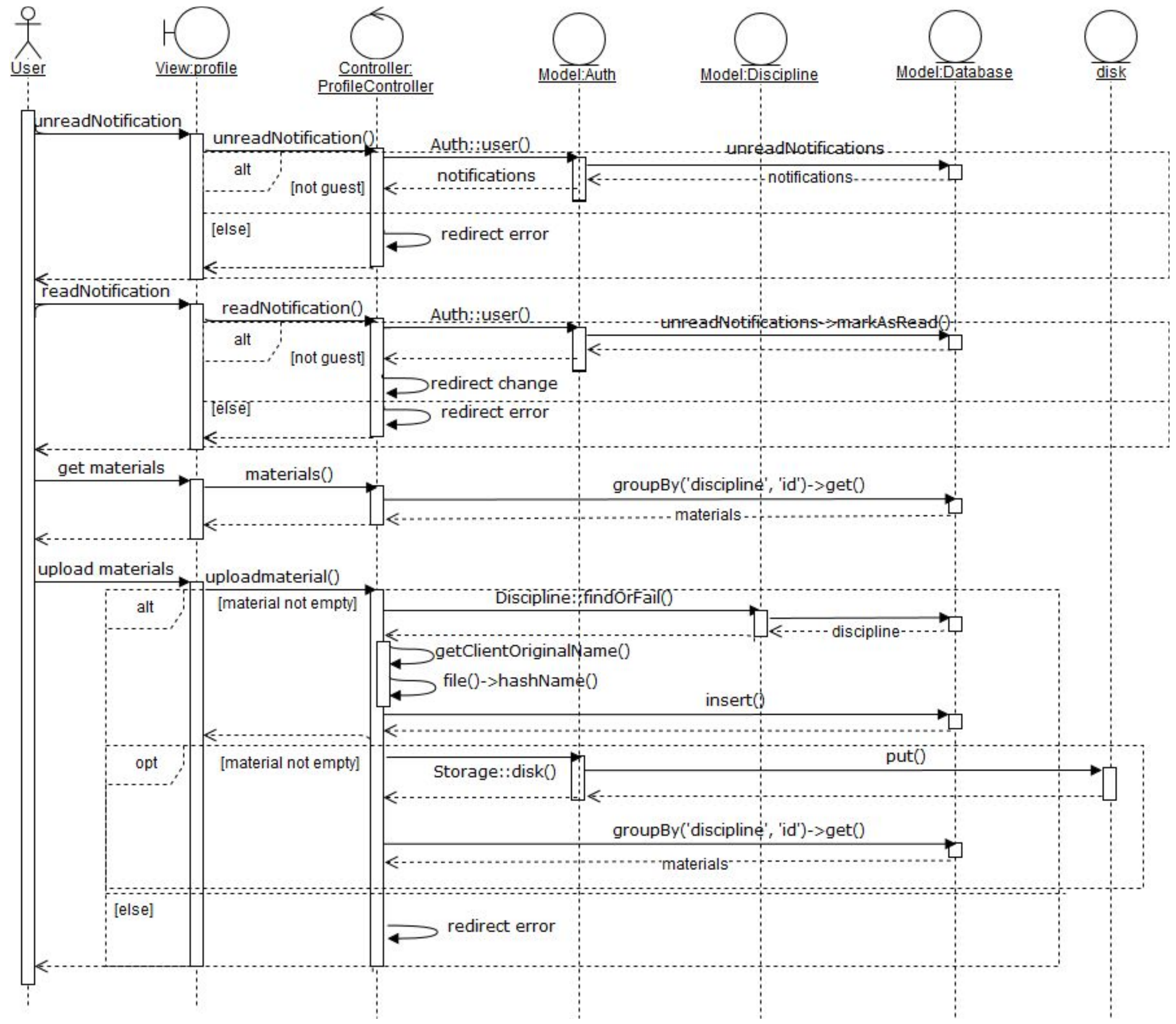
Model: Auth - for checking if guest or not

Discipline - find materials by discipline

CourseDetail - get all course related informations

View: profile - users can download course materials and receive notifications under profile

Controller: ProfileController - controller which control notification and materials operations (notification and material)l



<u>Event</u>

Model: Event - for CRUD(create, read, update, delete) operations in event table

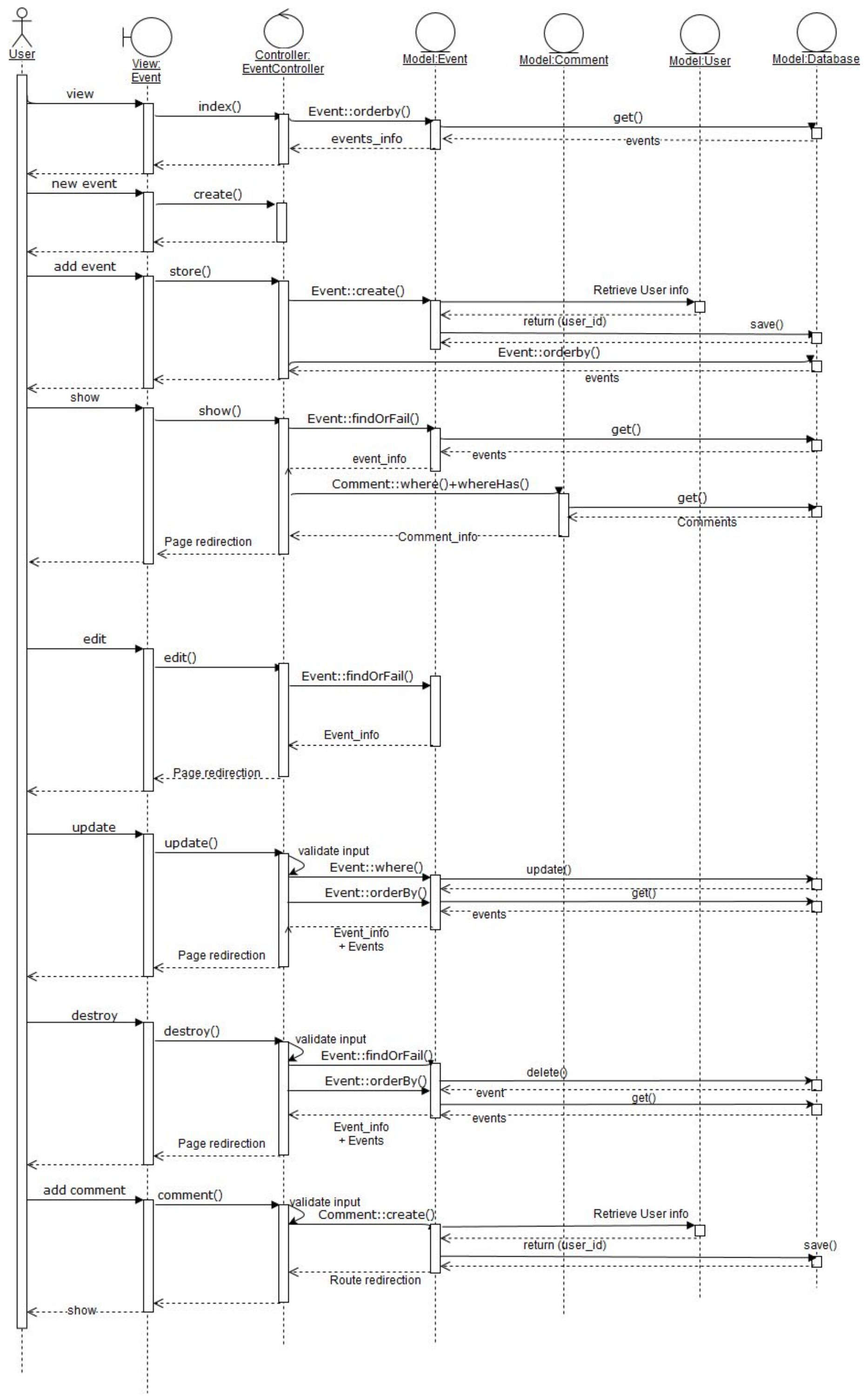
Comment - user can comment on event

User - for retrieving user's information

View: Event - we used RESTful resource controllers so there are event.index, event.create, event.store, event.show, event.edit, event.update, event.destroy routes for doing CRUD operations.

Controller: EventController - controller which implements the seven functions respectively in view.

(Event)



3.4 Limitations/Performance issues for MVC

Laravel is comparative slow and use more memory compare to other framework. As it uses a lot of symfony libraries which is slow, also for default CRUD operations, Laravel used "lazy loading" approach which means it only retrieve data when object is referenced somewhere, this may cause extra query. To optimize it, we can try to "eager loading" data so when call a query it immediately retrieve object for ready use using with()->get() instead of all() function calling of Eloquent model.

4.0 User interface design

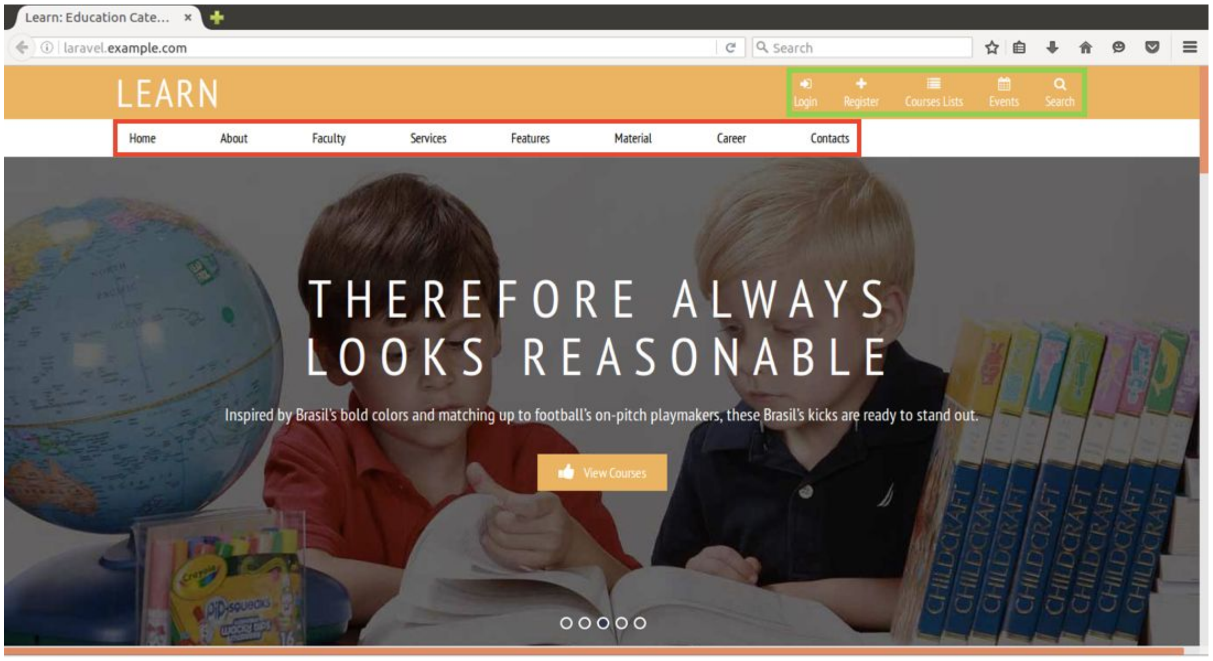
4.1 Description of the user interface

Homepage of eBoard will be shown to public at first having some important news and information such as class delay or course registration discount. Major functions which are registration, login and posts can be found at the top right hand side corner. Other useful functions are located in the middle of the website displayed by a bar providing contact, about us and course materials. Once the users click on these buttons, they can browse different web pages through hyperlink efficiently.

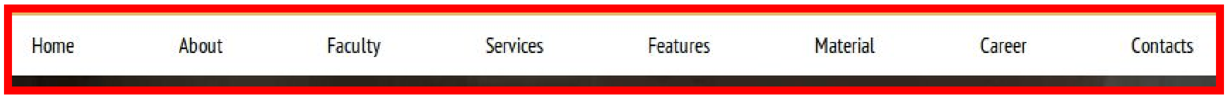
Registered users can additionally view and edit their user profiles at the top right hand side corner after finishing the login process as an authenticated user. Searching courses can be done by not only users, but also non-registered visitors. However, course registration is only available to users. Since the website will validate the user's identity during the user registration process, course registration management will become easier and effortless.

4.1.1 Screen images

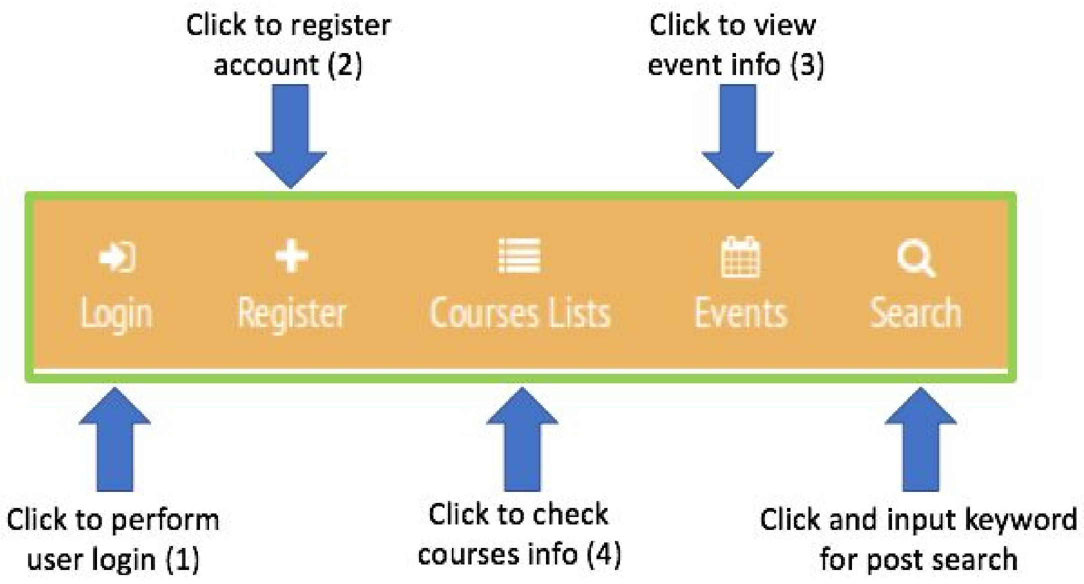
Homepage


Learn: Education Cate...
laravel.example.com
Search
LEARN
Login
Register
Courses Lists
Events
Search
Home
About
Faculty
Services
Features
Material
Career
Contacts
THEREFORE ALWAYS LOOKS REASONABLE
Inspired by Brasil's bold colors and matching up to football's on-pitch playmakers, these Brasil's kicks are ready to stand out.
View Courses

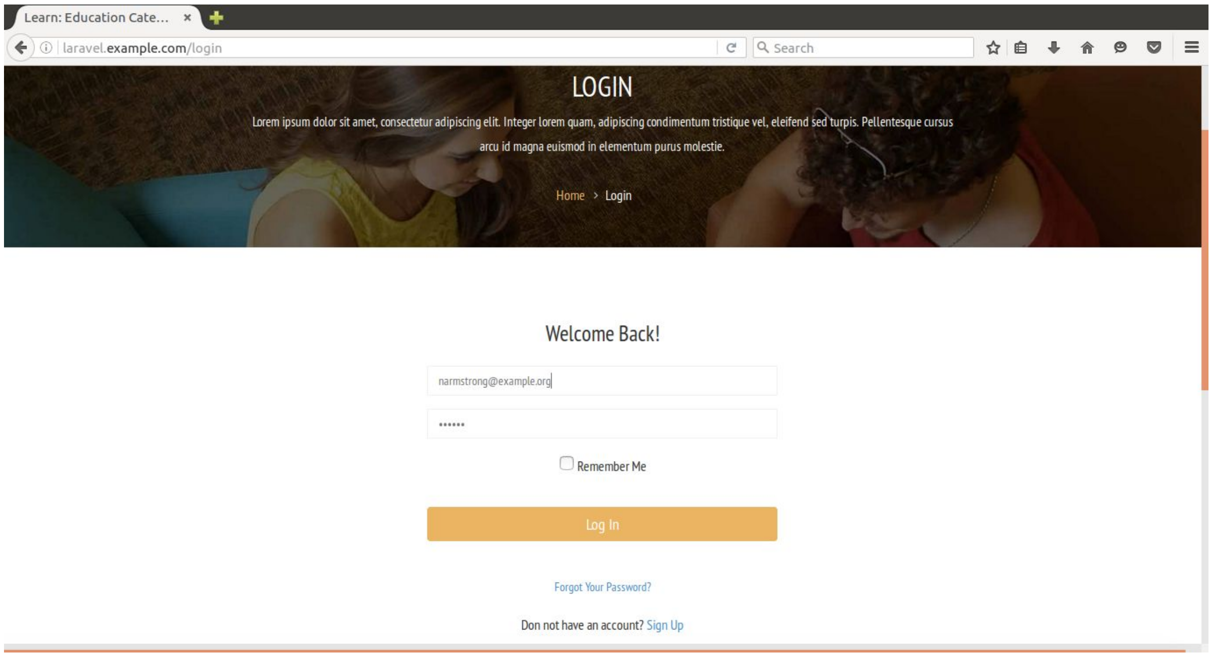
Information Bar (buttons)



Visitor Control Panel (buttons)


Click to register account (2)
Click to view event info (3)
Login
Register
Courses Lists
Events
Search
Click to perform user login (1)
Click to check courses info (4)
Click and input keyword for post search

Login Page



LOGIN

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer lorem quam, adipiscing condimentum tristique vel, eleifend sed turpis. Pellentesque cursus arcu id magna euismod in elementum purus molestie.

Home > Login

Welcome Back!

narmstrong@example.org

••••••

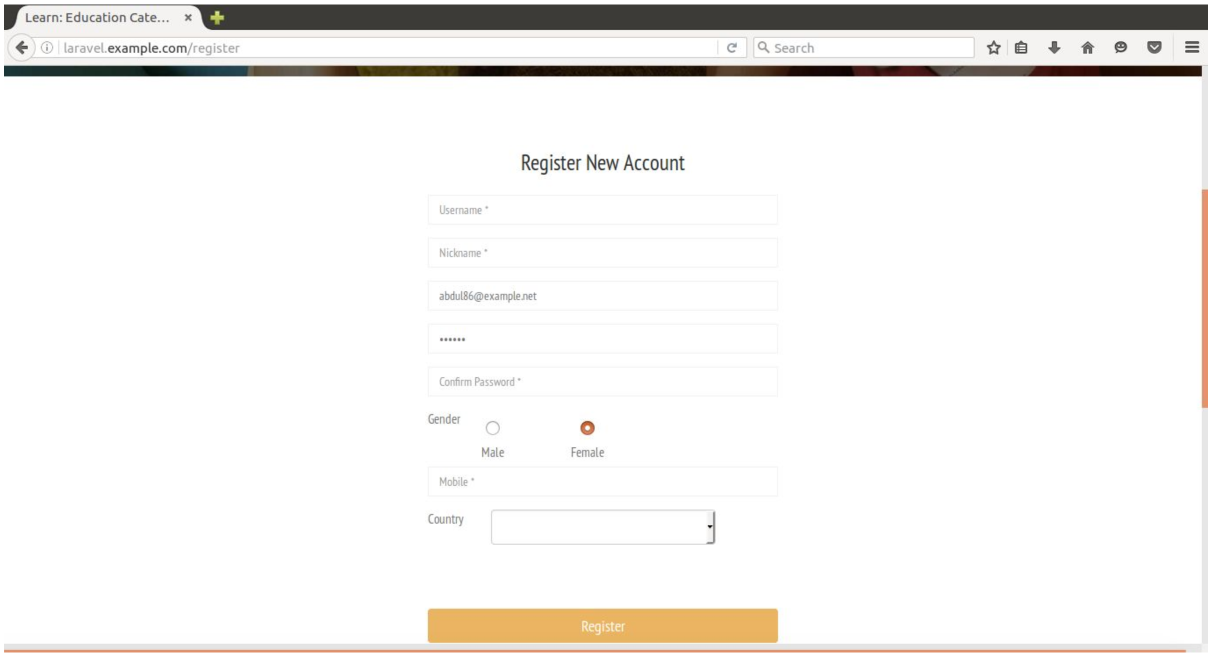
Remember Me

Log In

Forgot Your Password?

Don not have an account? Sign Up

Registration Page



Register New Account

Username *

Nickname *

abdul86@example.net

••••••

Confirm Password *

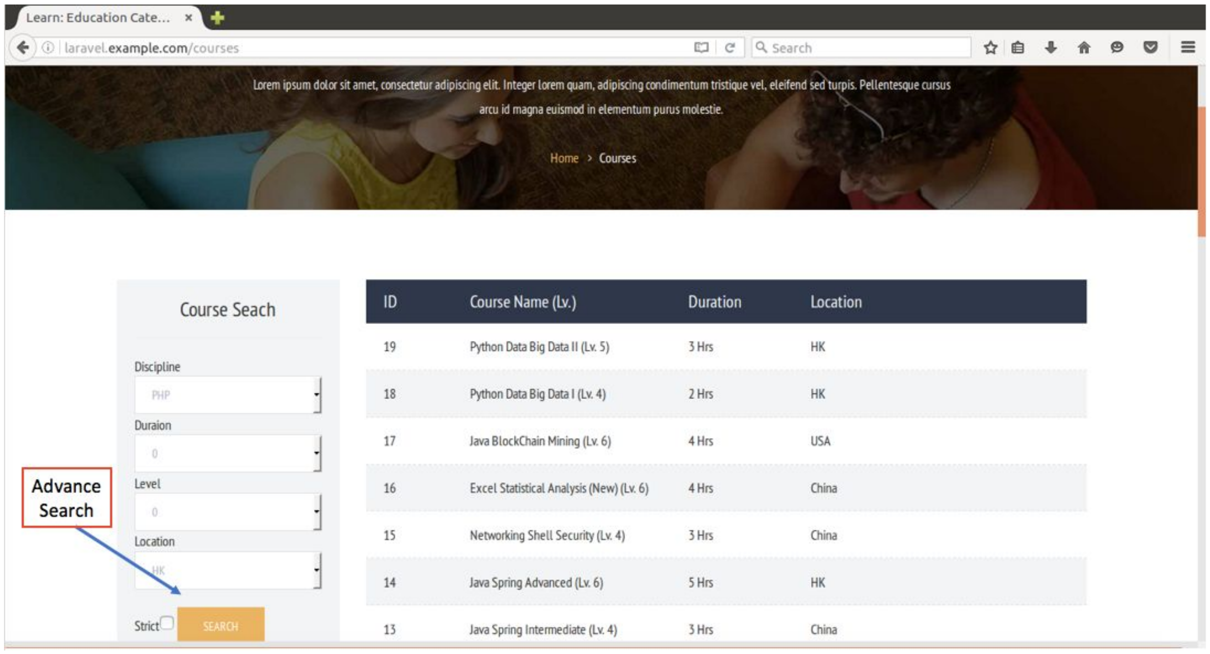
Gender Male [✓ Female]

Mobile *

Country

Register

Course Page



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer lorem quam, adipiscing condimentum tristique vel, eleifend sed turpis. Pellentesque cursus arcu id magna euismod in elementum purus molestie.

Home > Courses

Course Seach

Discipline: PHP

Duraion: 0

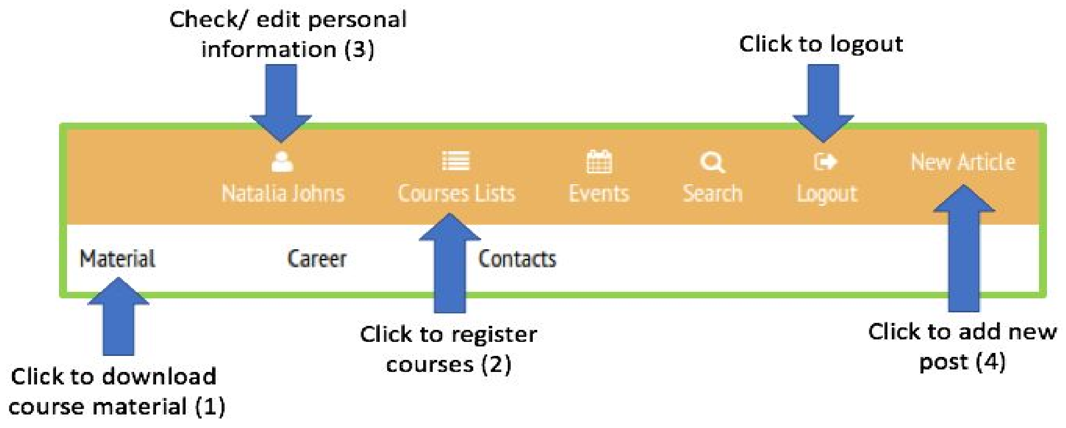
Level: 0

Location: HK

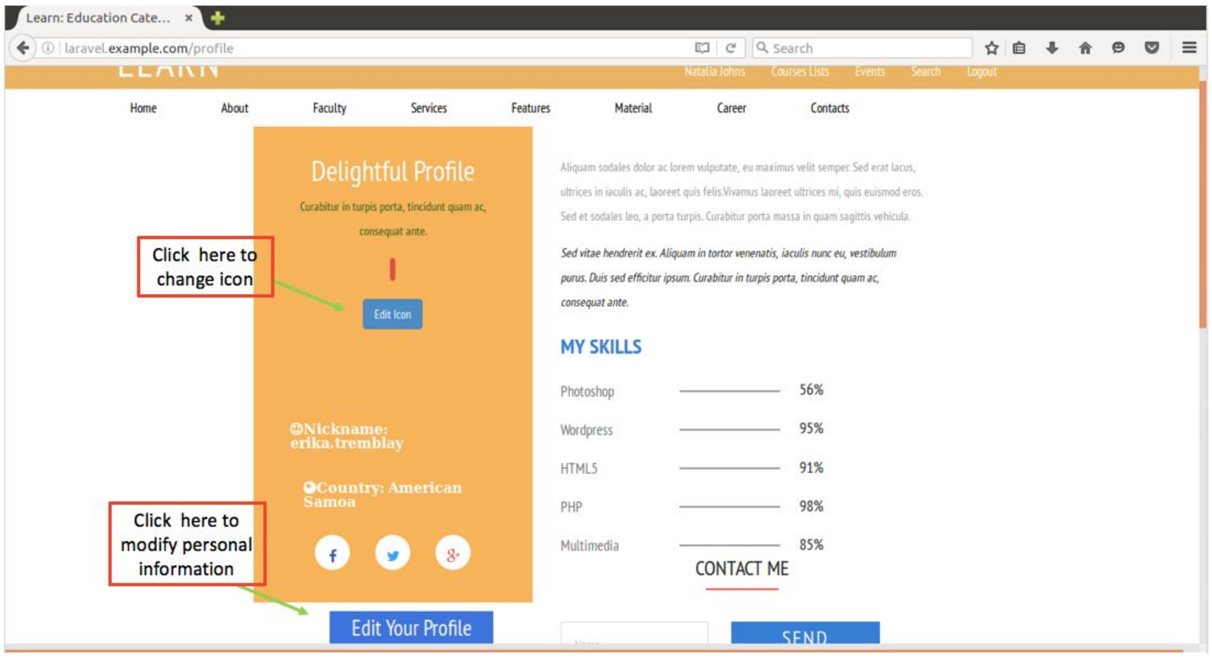
Strict ☐ SEARCH

ID	Course Name (Lv.)	Duration	Location
19	Python Data Big Data II (Lv. 5)	3 Hrs	HK
18	Python Data Big Data I (Lv. 4)	2 Hrs	HK
17	Java BlockChain Mining (Lv. 6)	4 Hrs	USA
16	Excel Statistical Analysis (New) (Lv. 6)	4 Hrs	China
15	Networking Shell Security (Lv. 4)	3 Hrs	China
14	Java Spring Advanced (Lv. 6)	5 Hrs	HK
13	Java Spring Intermediate (Lv. 4)	3 Hrs	China

User Control Panel (additional function)


Check/ edit personal information (3)
Click to logout
Natalia Johns
Courses Lists
Events
Search
Logout
New Article
Material
Career
Contacts
Click to download course material (1)
Click to register courses (2)
Click to add new post (4)

User Profile



Curabitur in turpis porta, tincidunt quam ac, consequat ante.

Nickname: erika.tremblay

Country: American Samoa

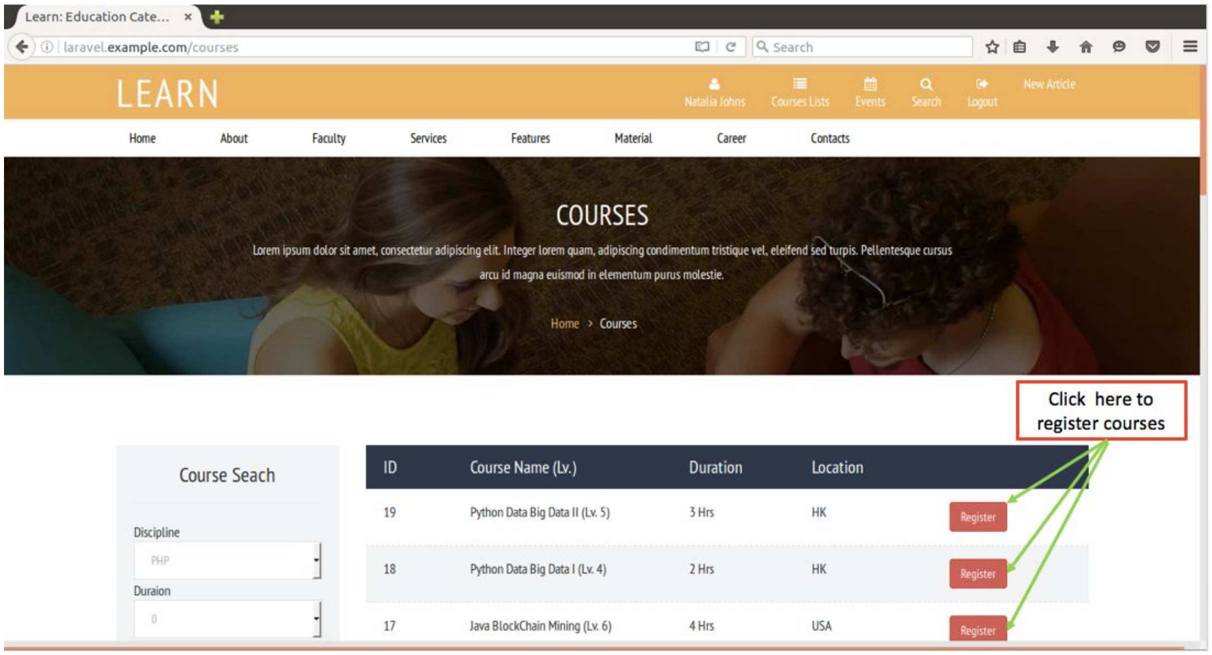
Aliquam sodales dolor ac lorem vulputate, eu maximus velit semper. Sed erat lacus, ultrices in iaculis ac, laoreet quis felis.Vivamus laoreet ultrices mi, quis euismod eros. Sed et sodales leo, a porta turpis. Curabitur porta massa in quam sagittis vehicula.

Sed vitae hendrerit ex. Aliquam in tortor venenatis, iaculis nunc eu, vestibulum purus. Duis sed efficitur ipsum. Curabitur in turpis porta, tincidunt quam ac, consequat ante.

MY SKILLS

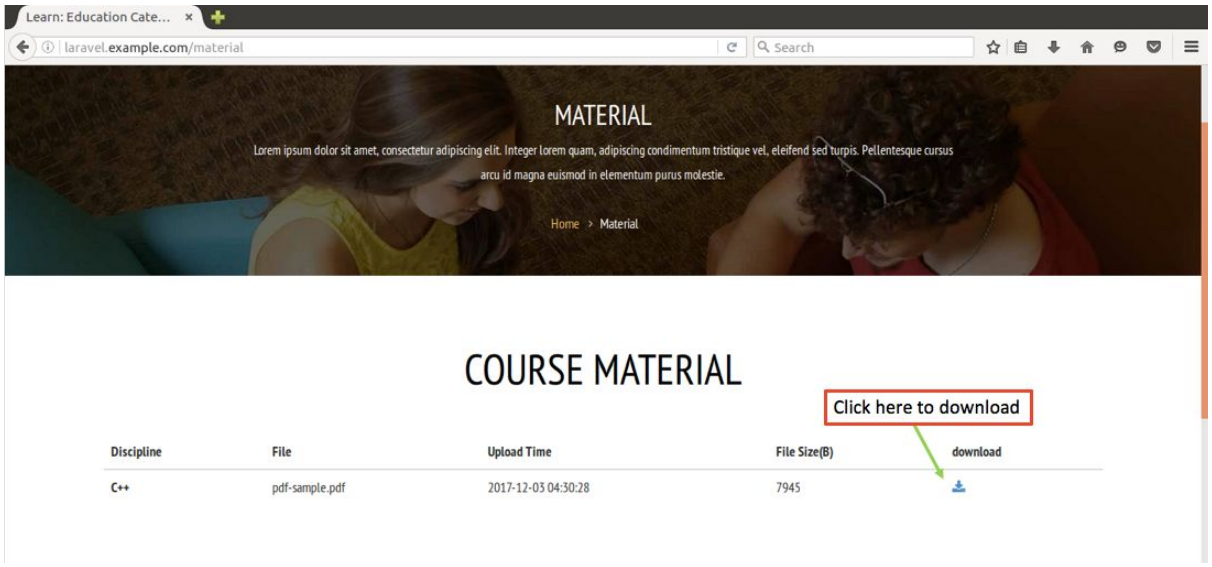
Photoshop	56%
Wordpress	95%
HTML5	91%
PHP	98%
Multimedia	85%

Course Registration Page


Learn: Education Cate...
laravel.example.com/courses
LEARN
Natalia Johns
Courses Lists
Events
Search
Logout
New Article
Home
About
Faculty
Services
Features
Material
Career
Contacts
COURSES
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer lorem quam, adipiscing condimentum tristique vel, eleifend sed turpis. Pellentesque cursus arcu id magna euismod in elementum purus molestie.
Home > Courses
Click here to register courses
Course Seach
Discipline
PHP
Duraion
0

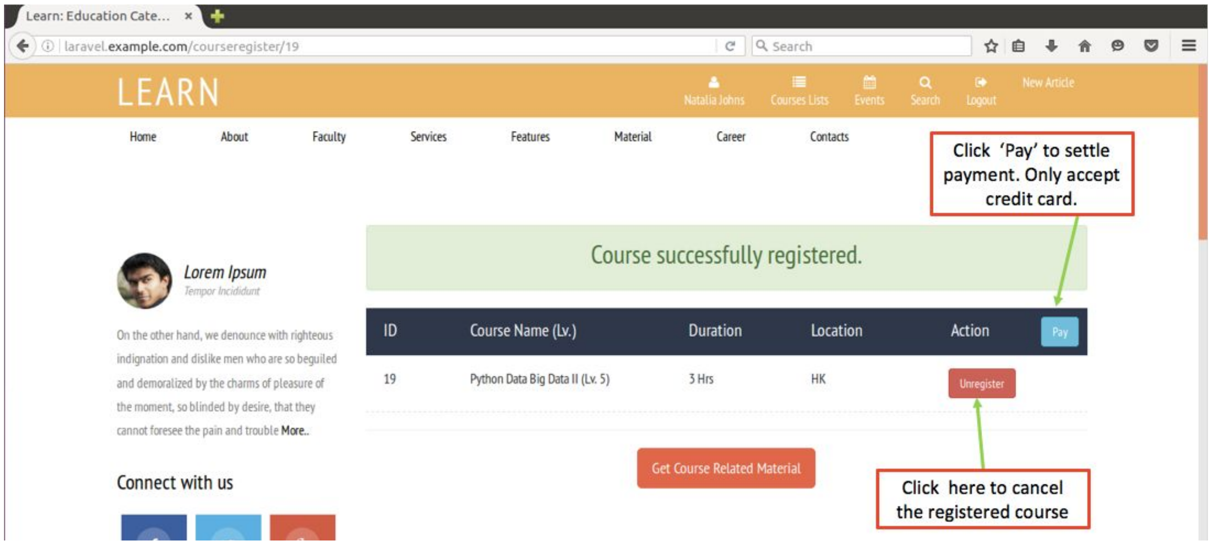
ID	Course Name (Lv.)	Duration	Location	
19	Python Data Big Data II (Lv. 5)	3 Hrs	HK	Register
18	Python Data Big Data I (Lv. 4)	2 Hrs	HK	Register
17	Java BlockChain Mining (Lv. 6)	4 Hrs	USA	Register

Course Materials


Learn: Education Cate...
laravel.example.com/material
MATERIAL
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer lorem quam, adipiscing condimentum tristique vel, eleifend sed turpis. Pellentesque cursus arcu id magna euismod in elementum purus molestie.
Home > Material
COURSE MATERIAL
Click here to download

Discipline	File	Upload Time	File Size(B)	download
C++	pdf-sample.pdf	2017-12-03 04:30:28	7945	⬇

My Course Page



Click 'Pay' to settle payment. Only accept credit card.

Course successfully registered.

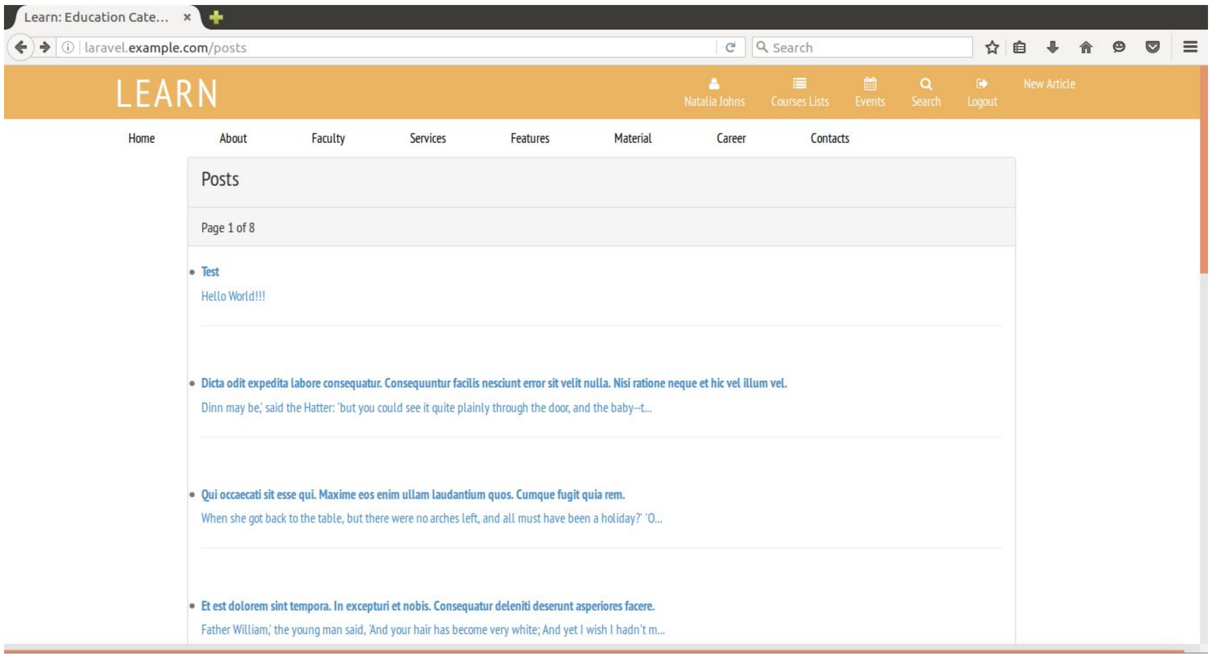
ID	Course Name (Lv.)	Duration	Location	Action
19	Python Data Big Data II (Lv. 5)	3 Hrs	HK	Unregister

Lorem Ipsum
Tempor Incididunt

On the other hand, we denounce with righteous indignation and dislike men who are so beguiled and demoralized by the charms of pleasure of the moment, so blinded by desire, that they cannot foresee the pain and trouble **More..**

Click here to cancel the registered course

Post List

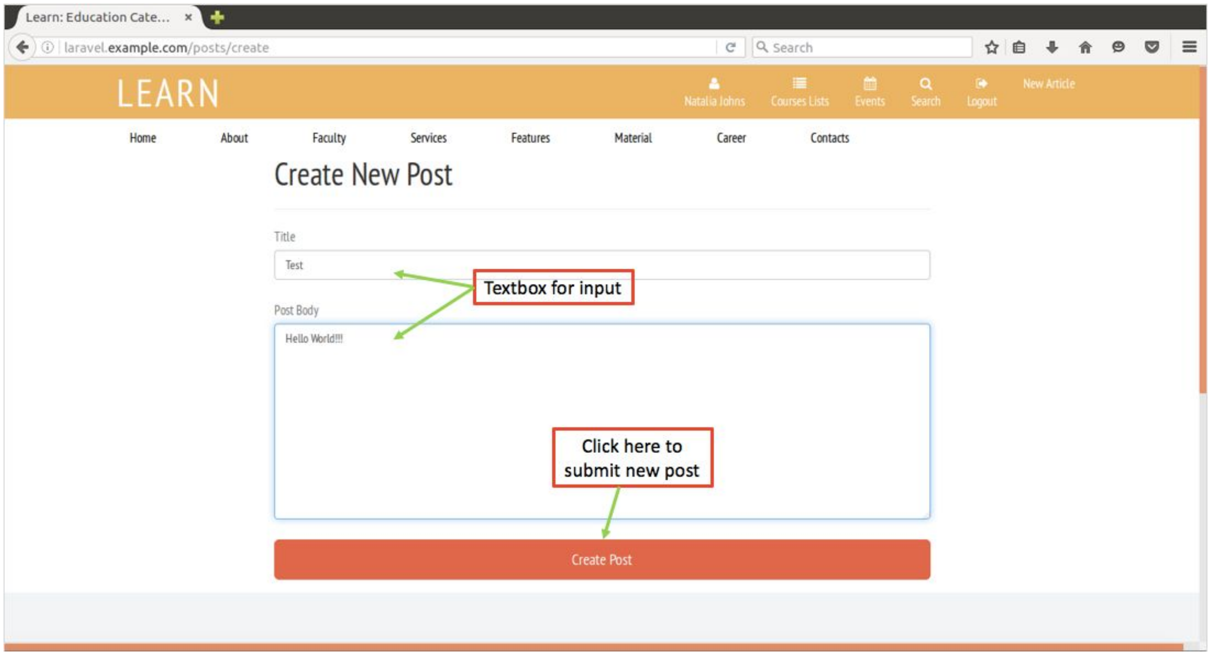


Posts

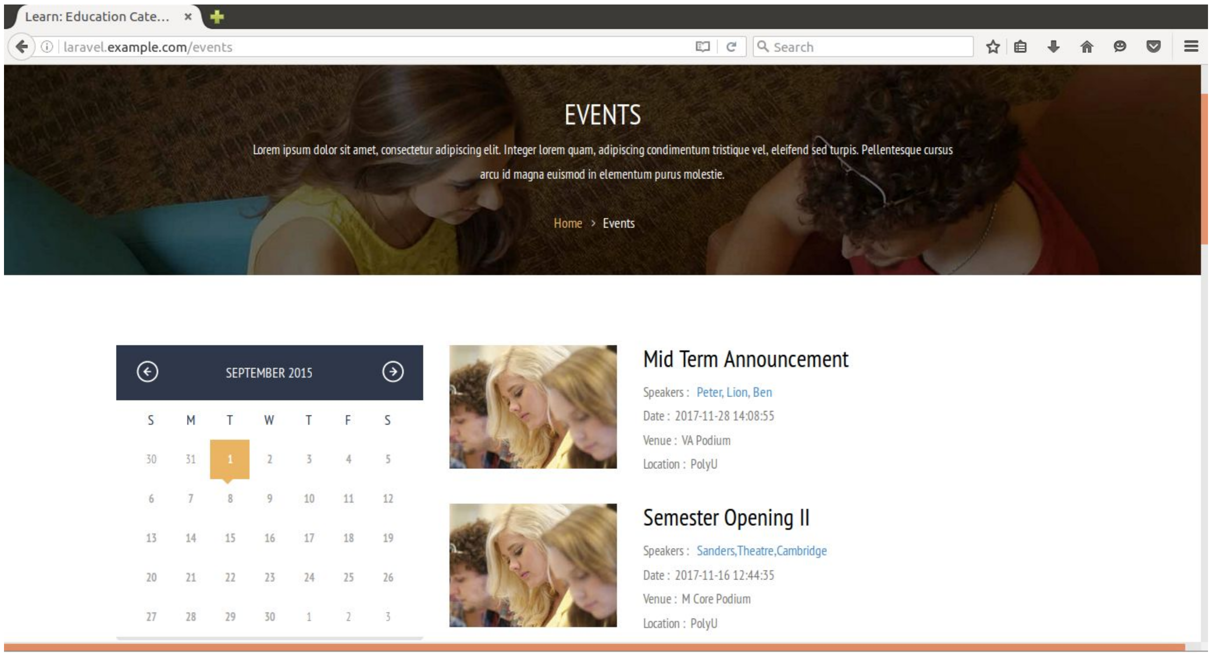
Page 1 of 8

- Test
  Hello World!!!
- Dicta odit expedita labore consequatur. Consequuntur facilis nesciunt error sit velit nulla. Nisi ratione neque et hic vel illum vel.
  Dinn may be,' said the Hatter: 'but you could see it quite plainly through the door, and the baby--t...
- Qui occaecati sit esse qui. Maxime eos enim ullam laudantium quos. Cumque fugit quia rem.
  When she got back to the table, but there were no arches left, and all must have been a holiday?' 'O...
- Et est dolorem sint tempora. In excepturi et nobis. Consequatur deleniti deserunt asperiores facere.
  Father William,' the young man said, 'And your hair has become very white; And yet I wish I hadn't m...

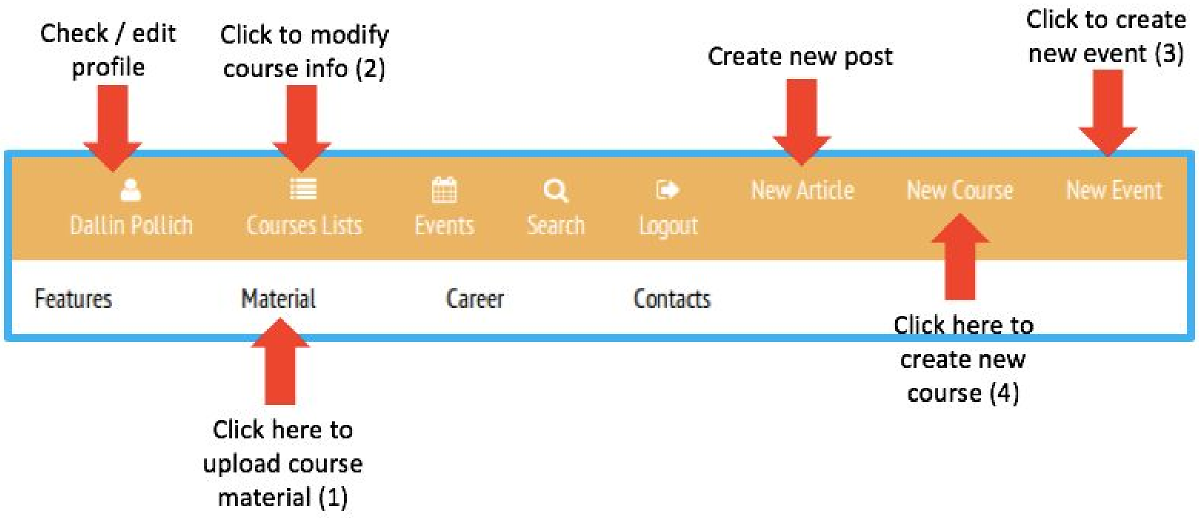
Create Post Page


Learn: Education Cate...
laravel.example.com/posts/create
LEARN
Natalia Johns
Courses Lists
Events
Search
Logout
New Article
Home
About
Faculty
Services
Features
Material
Career
Contacts
Create New Post
Title
Test
Textbox for input
Post Body
Hello World!!!
Click here to submit new post
Create Post

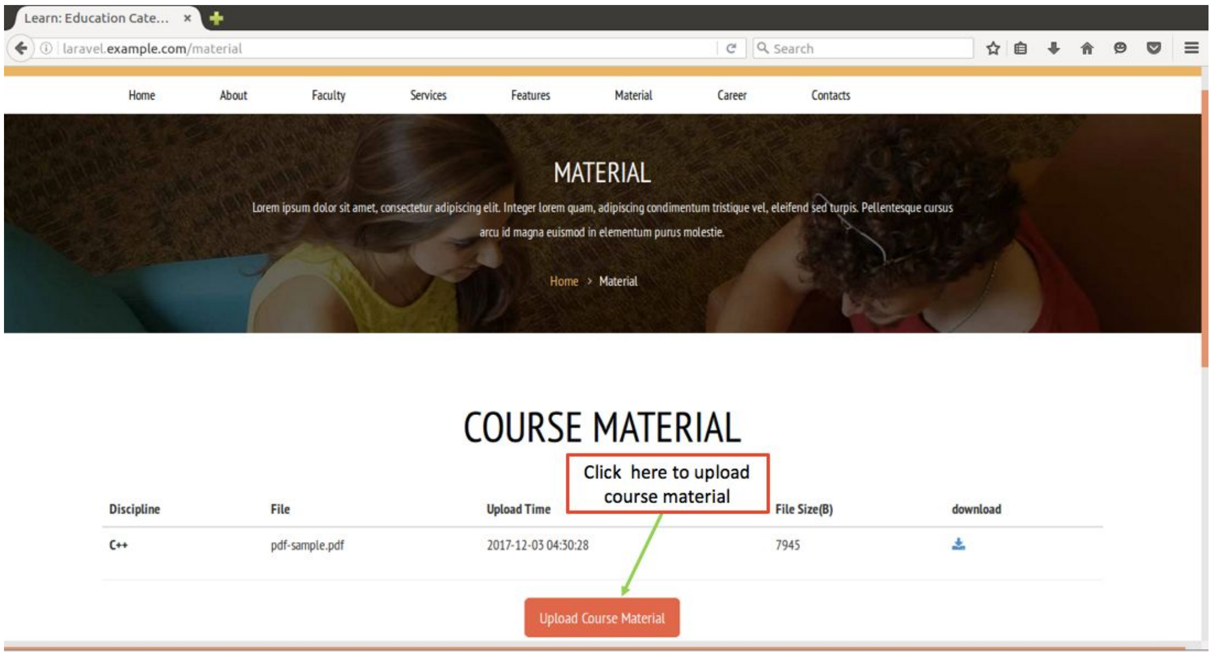
Event Page


Learn: Education Cate...
laravel.example.com/events
EVENTS
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer lorem quam, adipiscing condimentum tristique vel, eleifend sed turpis. Pellentesque cursus arcu id magna euismod in elementum purus molestie.
Home > Events
SEPTEMBER 2015
Mid Term Announcement
Speakers : Peter, Lion, Ben
Date : 2017-11-28 14:08:55
Venue : VA Podium
Location : PolyU
Semester Opening II
Speakers : Sanders,Theatre,Cambridge
Date : 2017-11-16 12:44:35
Venue : M Core Podium
Location : PolyU

Teacher Control Panel (additional functions)



Upload Course Materials

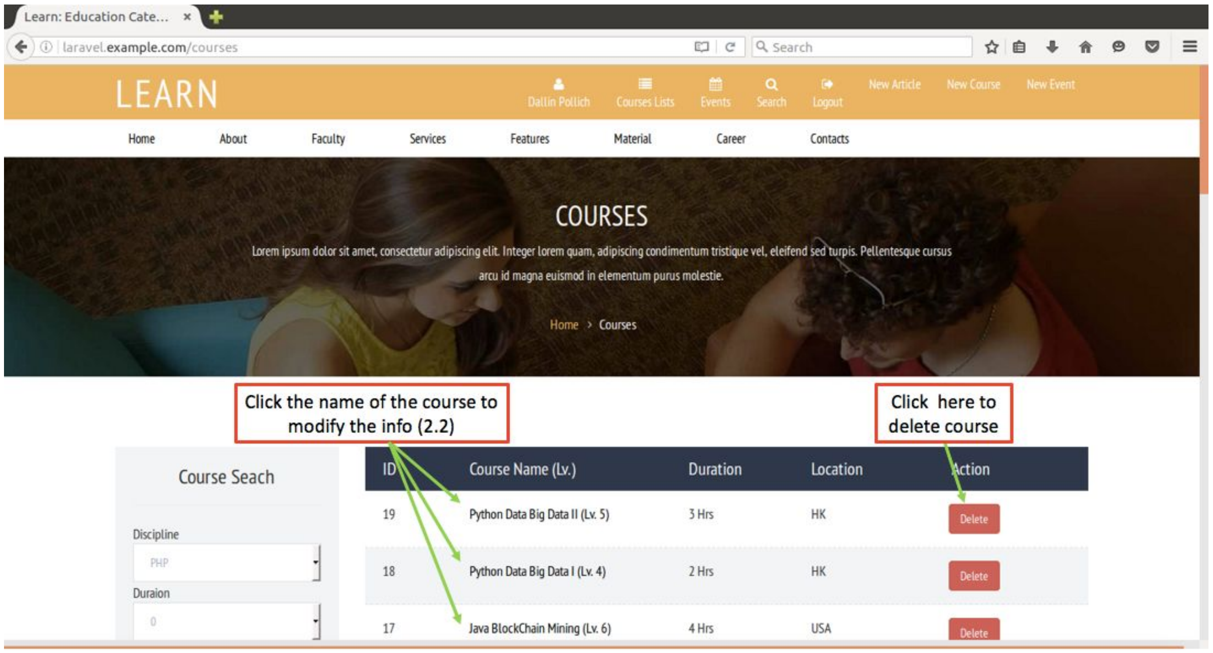


Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer lorem quam, adipiscing condimentum tristique vel, eleifend sed turpis. Pellentesque cursus arcu id magna euismod in elementum purus molestie.

COURSE MATERIAL

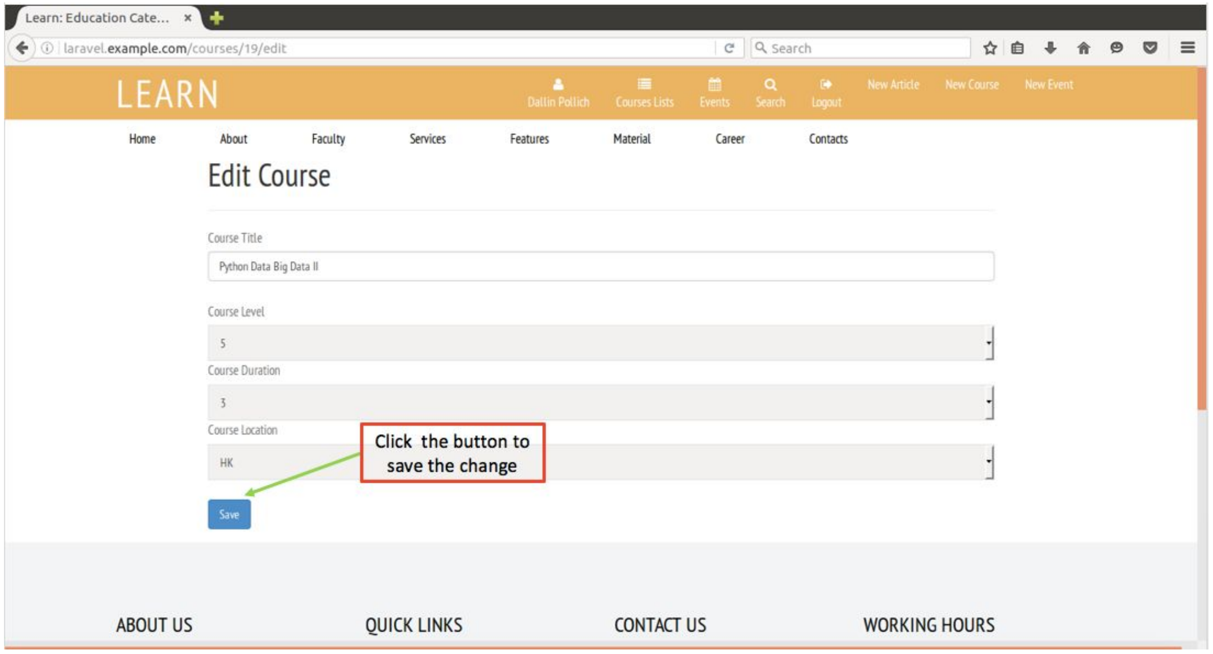
Discipline	File	Upload Time	File Size(B)	download
C++	pdf-sample.pdf	2017-12-03 04:30:28	7945	

Edit Course List


Learn: Education Cate...
laravel.example.com/courses
LEARN
Dallin Pollich
Courses Lists
Events
Search
Logout
New Article
New Course
New Event
Home
About
Faculty
Services
Features
Material
Career
Contacts
COURSES
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer lorem quam, adipiscing condimentum tristique vel, eleifend sed turpis. Pellentesque cursus arcu id magna euismod in elementum purus molestie.
Home > Courses
Click the name of the course to modify the info (2.2)
Click here to delete course
Course Seach
Discipline
PHP
Duraion
0

ID	Course Name (Lv.)	Duration	Location	Action
19	Python Data Big Data II (Lv. 5)	3 Hrs	HK	Delete
18	Python Data Big Data I (Lv. 4)	2 Hrs	HK	Delete
17	Java BlockChain Mining (Lv. 6)	4 Hrs	USA	Delete

Edit Courses



Edit Course

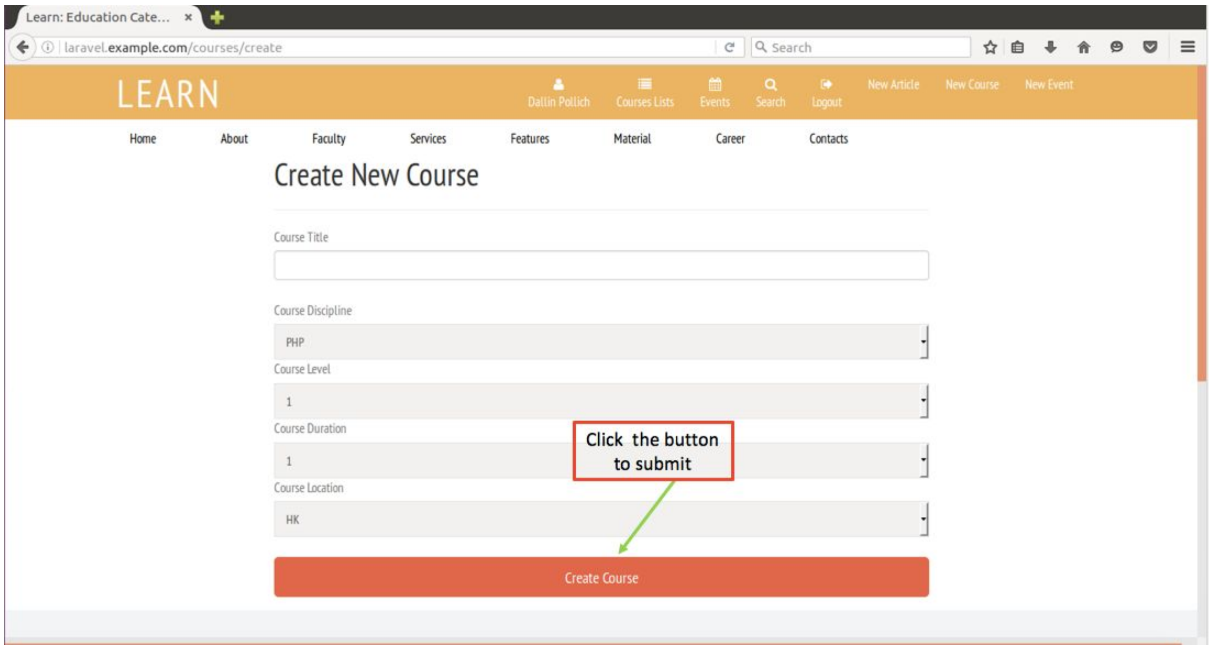
Course Title: Python Data Big Data II

Course Level: 5

Course Duration: 3

Course Location: HK

New Course



Create New Course

Course Title

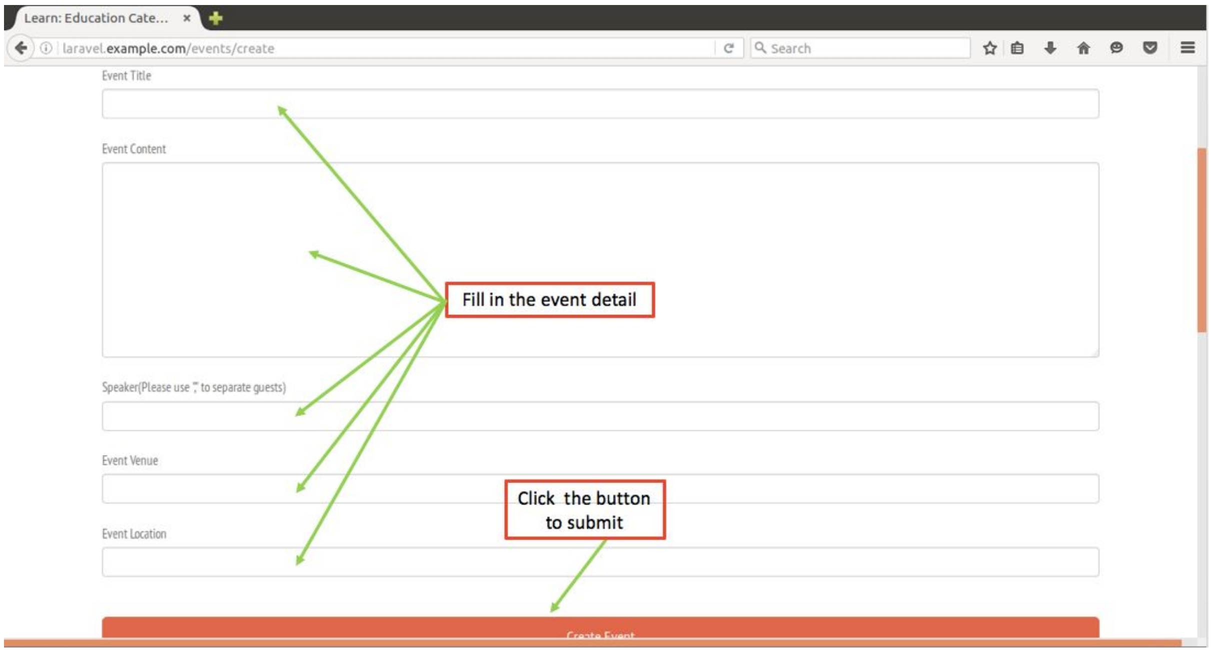
Course Discipline: PHP

Course Level: 1

Course Duration: 1

Course Location: HK

New Event



Event Title

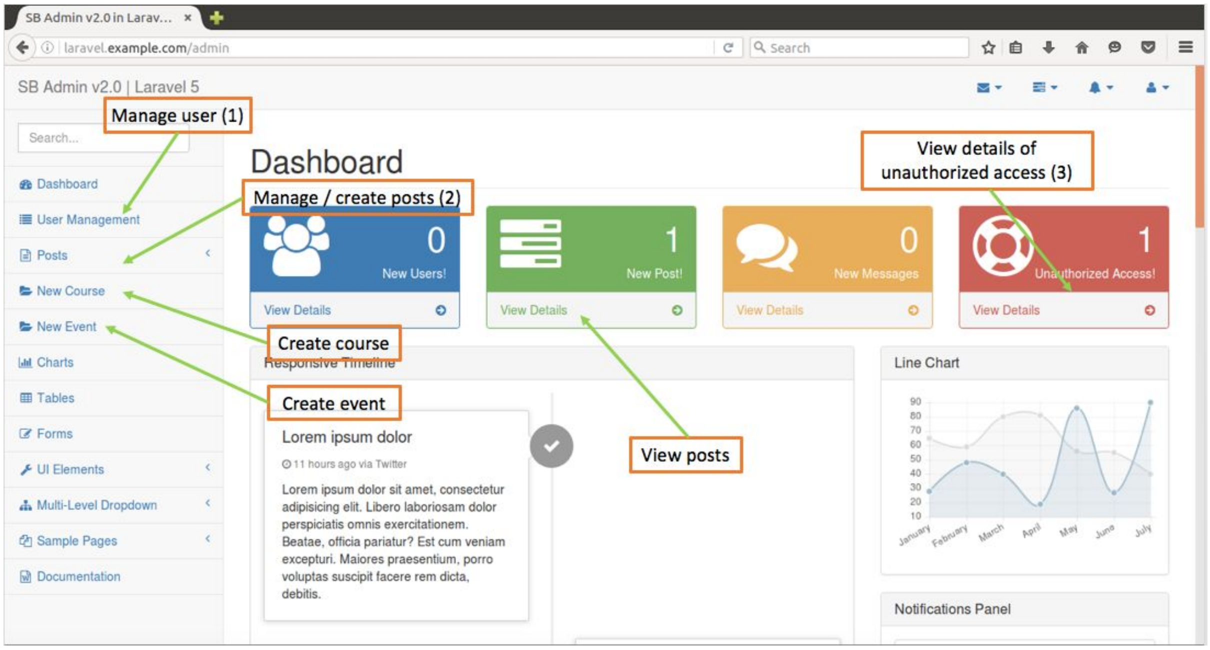
Event Content

Speaker(Please use ';' to separate guests)

Event Venue

Event Location

Admin Panel

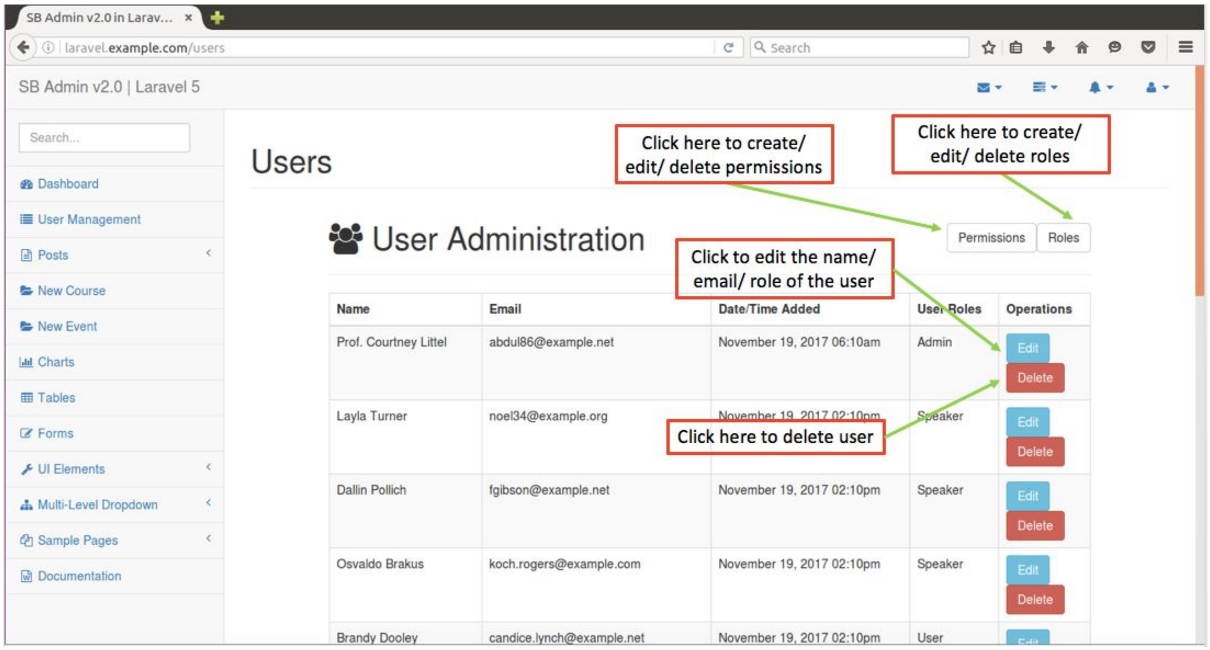


Lorem ipsum dolor

11 hours ago via Twitter

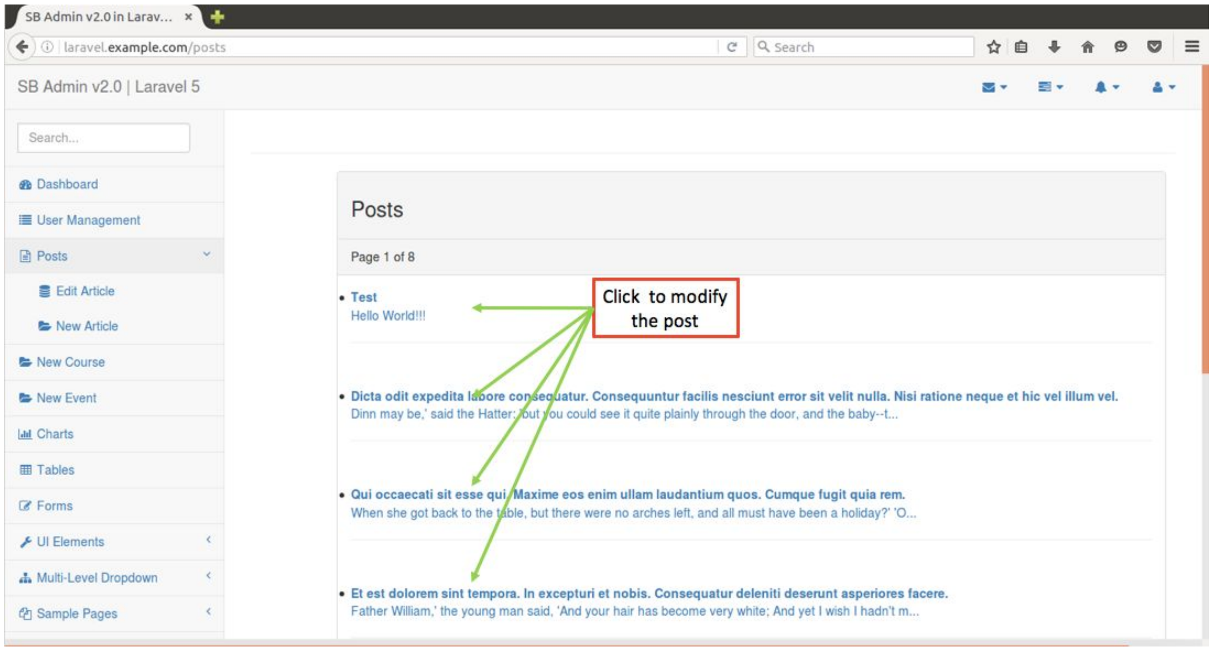
Lorem ipsum dolor sit amet, consectetur adipisicing elit. Libero laboriosam dolor perspiciatis omnis exercitationem. Beatae, officia pariatur? Est cum veniam excepturi. Maiores praesentium, porro voluptas suscipit facere rem dicta, debitis.

User Management


SB Admin v2.0 in Larav...
laravel.example.com/users
Search
SB Admin v2.0 | Laravel 5
Search...
Dashboard
User Management
Posts
New Course
New Event
Charts
Tables
Forms
UI Elements
Multi-Level Dropdown
Sample Pages
Documentation
Users
Click here to create/ edit/ delete permissions
Click here to create/ edit/ delete roles
User Administration
Permissions
Roles
Click to edit the name/ email/ role of the user
Click here to delete user

Name	Email	Date/Time Added	User Roles	Operations
Prof. Courtney Littel	abdul86@example.net	November 19, 2017 06:10am	Admin	Edit Delete
Layla Turner	noel34@example.org	November 19, 2017 02:10pm	Speaker	Edit Delete
Dallin Pollich	fgibson@example.net	November 19, 2017 02:10pm	Speaker	Edit Delete
Osvaldo Brakus	koch.rogers@example.com	November 19, 2017 02:10pm	Speaker	Edit Delete
Brandy Dooley	candice.lynch@example.net	November 19, 2017 02:10pm	User	Edit

Post Management



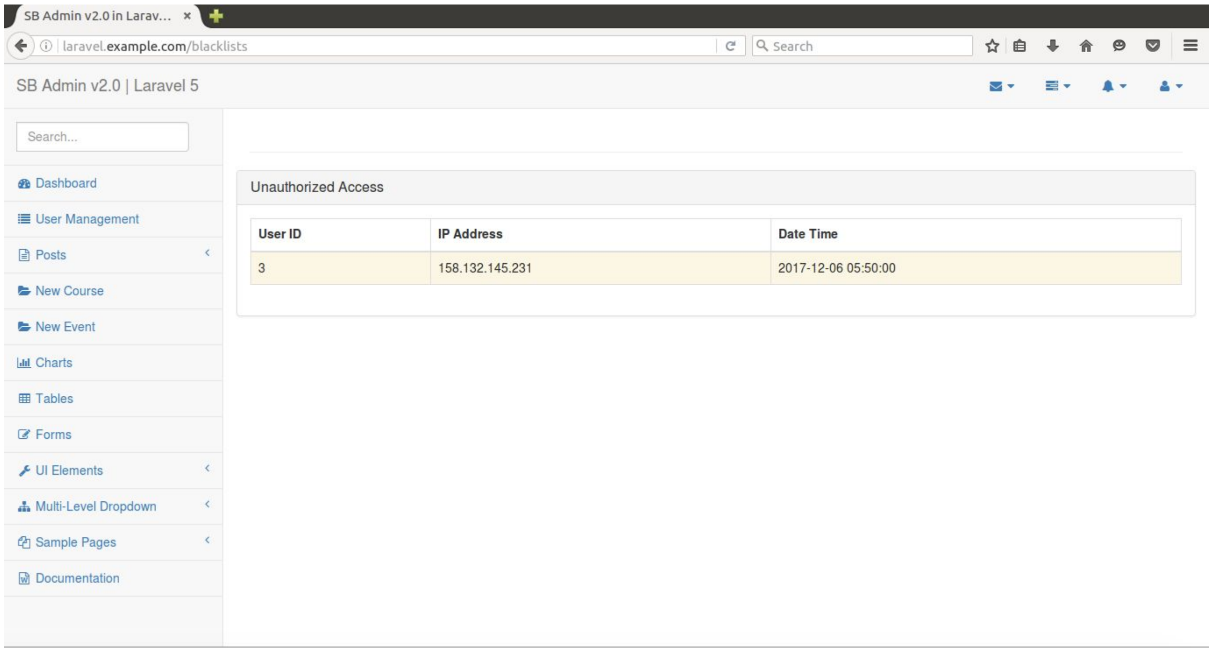
Posts

Page 1 of 8

Click to modify the post

- Test
  Hello World!!!
- Dicta odit expedita labore consequatur. Consequuntur facilis nesciunt error sit velit nulla. Nisi ratione neque et hic vel illum vel.
  Dinn may be,' said the Hatter; 'but you could see it quite plainly through the door, and the baby--t...
- Qui occaecati sit esse qui. Maxime eos enim ullam laudantium quos. Cumque fugit quia rem.
  When she got back to the table, but there were no arches left, and all must have been a holiday?' 'O...
- Et est dolorem sint tempora. In excepturi et nobis. Consequatur deleniti deserunt asperiores facere.
  Father William,' the young man said, 'And your hair has become very white; And yet I wish I hadn't m...

Unauthorized Access Details



Unauthorized Access

User ID	IP Address	Date Time
3	158.132.145.231	2017-12-06 05:50:00

4.1.2 Objects and actions

Role	Object and Action
Visitor	- Access to eBoard

	- Search courses (select options in drop-down menu) - View Event and Post pages to view them (related buttons) - Search Posts (enter related keywords) - User Registration (input valid information) - User Login (correct username and password)
User	- **All authorities of Visitor** - Logout (logout button) - View / Edit user profile (user profile button) - Course registration (register button in course list page) - Pay course fee (pay button in my course page) - Create posts (create button in create post page) - Download course materials (download button in material page)
Teacher	**- All authorities of User** **-** Edit course list (edit button in course list page) - Create new courses (new course button) - Create new event (new event button) - Upload course material (upload button in course materials page)
Admin	**- All authorities of Teacher** - Access to **admin panel** with the admin account - Website data analysis - User admission management (edit / delete buttons) - Post management (click on the title for modification) - Unauthorized access details (related access information list)

4.2 Interface design rules

The interface should be user-friendly and secure to protect users' information and the system.
- Website elements are well distributed and clear
- Big enough font size
- Comfortable color for display
- Interface should be different between visitor, user, teach and admin for identification
- Interface of the admin page should be redirected for security issue

4.3 Components available

Most used GUI components :
- Buttons row groups buttons in a set (Information Bar in the homepage / Control Panel at the top right hand side corner)
- Colored button illustrates the importance of the buttons (e.g. Edit / Delete)
- Slide-down motion saves space and is elegant (post search button)
- Drop-down menu saves space and provides options (search function for Course List)
- Text field handle users' input (e.g. User Registration / Keyword Search)

4.4 UIDS description

Above screen images show the UIDs. UIDs have the similar interface design but slightly different to the interface among visitor, user and teacher. Especially, admin panel is a completely distinct interface.

Role	Content (Top right hand side corner)
Visitor	- Login / Register / Course Lists / Events / Search - Course Register button is in color grey and unclickable (inside the Course List page)
User	- User Profile (named in user's nickname) / Course Lists / Events / Search / Logout / New Article (publish posts) - Course Register button is in color blue and clickable (inside the Course List page) - A new Download Course Materials button is available for download materials (inside the Course Materials page)
Teacher	- User Profile (named in teacher's nickname) / Course Lists / Events / Search / Logout / New Article (publish posts) / New Course / New Event - Severals new Edit buttons are available for edit Course Lists / Posts and Events (inside corresponding pages) - A new Upload Course Materials button is available for download materials (inside the Course Materials page)
Admin	**- No longer different** - Sidebar is provided : Dashboard / User Management / New Course / New Event / Charts / Tables / Forms / UI Elements / Multi-level Dropdown / Sample Pages / Documentation - New information will be shown to admin (e.g. website data analysis / User Detail / Post List / Unauthorized Access etc.)

5.0 Restrictions, limitations, and constraints

Performance / Behaviour Issues
eBoard is a course registration website supported for most web browser using php and html combining with database service. Display problems may be occurred in mobile or tablet views. For instance, the ratio may be distorted and dissatisfy users. Simple CSS styling is used to design the website

If the system becomes large enough, its structure will be complicated due to the great variety of functions of the website. Development team will feel confused and frustrated. Bugs and errors may occur frequently in the system. Fault location is difficult and waste much time. It will slow down the whole development process. Undetected bugs can exist inside the final product. Experienced staffs can give advices for the system to avoid bugs and.

Dynamic changes on requirements
More functionalities should be added into eBoard such as discussion or a stronger searching function in the future. In order to maintain the value of website, requirement specification shall be stated clear. The engineering teams need to cooperate with the system and end users to prepare the changes or predict the future requirements.

Component issue
Online resources and components are useful and shorten the development time. However, packing different these components together is difficult and it may crash the whole system. The new version of buy-in component may not meet the real needs of customers.Technical team should choose the components carefully or even build up a own system. It enhances the security level of the website too.

Attack from Hackers
Since database is used to stored the private information of users, database may be attacked by the hackers and cause information leak. Temporarily, eBoard can protected from basic SQL injection and account brute force attack by adding quotes to user inputs. But DDoS is hard to be distinguished. In order to protect users' information, a secure database scheme should be established. The security scheme should provide a stable environment for the website. For instance, eBoard can run smoothly even though attackers are trying to take DDoS attack to the web server.

6.0 Testing Issues

6.1 Classes of tests

CSS Test

We basically test the css elements whether elements are display properly or not by observation.

1. Is homepage and other page display normally as user want?
2. Any ratio distortion and location problems of css elements? (accidently moved?)

HTML Object Test

Try the HTML objects whether they work fine or not.

1. Do the click on picture successfully link redirect user to pages ? (e.g. Homepage)
2. Do the click on hyperlink successfully link redirect user to pages ? (e.g. Event details)
3. Do button actions successfully link to pages or pass the value? (e.g.Submit)
4. Do drop-down menu successfully pass the value? (e.g. Course search)
5. Do text field successfully pass the value? (e.g. Keyword search on Posts)

SQL Actions

Try the SQL query whether successful query is made or not.

1. Try user registration and login that recently register accounts.
2. Try course registration and check My Course for its history.
3. Try search for courses or posts and see the results.

Component Connection

Try the connection and performance of the website as well as the database

1. Do the connection between the platform and database is valid?
2. Does the performance of the website result in major degrade when a lot of request coming in?
3. The data transmitted should well encrypted that network eavesdropping cannot directly reveal the data.

MVC action

Try the data flow of "model to view", view to controller" and "controller to model"

1. Does null value request properly handled?
2. Does model being saved successfully?
3. Does the view receive the correct model?

6.2 Expected software response

<u>CSS Test</u>

All CSS result should follow the the UI screen captures.
1. Web pages should display properly in a clear manner.
2. Elements should be group cleanly together without moving to wrong places or overlapping with each other.

<u>HTML Object Test</u>

1. Click on the homepage picture should redirect user to the homepage.
2. Click on the hyperlink of "Open Ceremony" should redirect user to that event page for more details (Time, Venue, Tutors etc.)
3. "Submit" button of user registration should pass data to other pages. (HTML GET method to check)
4. Drop-down menu should pass the value such as "PHP", "Level 1" for Course Search. (HTML GET method to check)
5. Text field should pass the text inputs such as for "Peter", "male", "peter123@gmail.com" for User Registration. (HTML GET method to check)

<u>SQL Actions</u>

1."Submit" button of user registration should pass the input to the database. If inputs are valid and non-duplicated, visitor can login as an user using that recent account.
2. Course registration history should added to related user' My Course.
3. Input Discipline="PHP" ; Level="1" should select and display all PHP Level 1 courses.

<u>Component Connection</u>

1. SQL query should be able to made (e.g. SELECT * FROM User;)
2. SQL query result should be shown within 3s for no significant delay.
3. Important data directly select from the database should be hashed. (e.g. hexadecimal string when SELECT user_pw FROM USER;)

<u>MVC action</u>

1. NULL value request should not any error and an error notification is made to users.
2. Modification can be shown when model is changed. (e.g. Visitor can use recently registered account to login as an user in eBoard)
3. For Course Search, only "JAVA Level 3" course should be shown after selecting "JAVA" for discipline and "3" for level in the drop-down menu of Course Search.

6.3 Performance bounds

Assuming the software can support approximately 80000 visitors/users which will generate 4500 customer interaction as outlined in the table below:

No	Description	Pages	Percentage
1	Portal	Login, Logout	40%
2	Transaction	payment, stripe portal	5%
3	Profile edit	myprofile	5%
4	Course register	course(index, register, unregister)	5%
5	Course Search	course search	10%
6	Material Search	material (search, download)	10%
7	Notification reader	notification (course amendment, news)	15%
8	Post issue	post(index, create, search)	10%

6.4 Identification of critical components

Model, View and Controller (MVC) will be the critical components since these three structure are the main architectural design of Laravel which is a MVC framework. Most portion of request, data flow and model action will locate inside these three components.

MVC is used to manipulate the process in database. In the future business environment, stakeholder and end users shall request for more functionality such as chatroom, online tasks. New attributes,operations and data models are needed to insert to eBoard. System will change a lot.

Moreover, program languages and system environment change a lot. The system value is also declining year by year. MVC maintenances are needed to ensure business daily services and maintain its business value.