

អារម្ភកថា

សៀវភៅនេះត្រូវបានចងក្រងឡើងក្នុងគោលបំណងចាត់ទុកជាកសារសម្រាប់បង្ហាញពីកិច្ចសិក្សា ស្រាវជ្រាវលើ Web Development របស់ក្រុមការងារដែលសិក្សានៅសាកលវិទ្យាល័យភូមិន្ទភ្នំពេញ ផ្នែក ព័ត៌មានវិទ្យា ជំនាន់ទី 17 ឆ្នាំ 4 ថ្នាក់ E5 ។ ក្រុមយើងខ្ញុំ បានសម្រេចចិត្តជ្រើសរើសនូវការបង្កើតនូវ ប្រព័ន្ធ CMS (Content Management System) ដែលនឹងធ្វើការគ្រប់គ្រងទៅលើផ្នែក Article Management ដែលយើងបានហៅវាថា “ប្រជុំអត្ថបទ CMS”។ ប្រព័ន្ធនេះបង្កើតឡើងដើម្បីធ្វើការបង្កើតនូវ គេហទំព័រសំរាប់ចែកចាយអត្ថបទ ជាព័ត៌មាន ឬក៏ជាចំនេះដឹងសំរាប់ចែកចាយទៅអោយអ្នកប្រើប្រាស់។ ប្រព័ន្ធនេះផងដែរបានបង្កើតគេហទំព័រជាលក្ខណៈ Dynamic ដែលមានរួមទាំង Back-End និង Front-End។ ការងារនេះអាចធ្វើទៅបានដោយសារមានការជ្រោមជ្រែងយ៉ាងយកចិត្តទុកដាក់ពីសំណាក់លោក សាស្ត្រាចារ្យ ជឹម ប៊ុនធឿន និងការខិតខំប្រឹងប្រែងពីសំណាក់សមាជិកក្រុមទាំងមូលផងដែរ។

ថ្វីត្បិតតែមានការខិតខំស្រាវជ្រាវយ៉ាងណាក៏ដោយក៏នៅតែមានកំហុសមួយចំនួនដោយចៀសមិនបាន ដែលនាំអោយ ប្រព័ន្ធ CMS របស់យើងនៅមានកង្វះខាតមួយចំនួនដែលត្រូវការអភិវឌ្ឍន៍បន្ថែម នៅពេល អនាគត។ ដូចនេះយើងសូមអភ័យទោសទុកជាមុនរាល់កំហុសឆ្គងដែលកើតឡើងដោយអចេតនានៅក្នុង ប្រព័ន្ធរបស់យើង ហើយយើងក៏រង់ចាំទទួលស្វាគមន៍នូវគំនិតយោបល់ រិះគន់ស្ថាបនាសម្រាប់អភិវឌ្ឍន៍ ប្រព័ន្ធ CMS នេះអោយកាន់តែរីកចម្រើនល្អជាលំដាប់។

ភ្នំពេញ, ថ្ងៃទី ០៥ ខែមិនា ឆ្នាំ ២០១៧

សូមអរគុណ!

មាតិកា

សេចក្តីផ្តើម	3
១. គោលបំណង	4
២. រចនាសម្ព័ន្ធនៃការបង្កើតប្រព័ន្ធ	4
២.១. ទំហំនៃប្រព័ន្ធ	4
២.២. Use-Case Diagram	5
២.៣. Database Design	6
២.៤. Interface Design	7
២.៤.១. Back-End	8
២.៤.២. Front-End.....	18
២.៥. API Document	20
៣. Implementation.....	26
៣.១. Application Flow	26
៣.២. បច្ចេកវិទ្យាប្រើប្រាស់.....	28
៤. ការពិភាក្សា.....	30
៤.១. ចំណុចខ្លាំង ចំណុចខ្សោយ	30
៤.២. គំរោងអនាគត	30
៥. សេចក្តីបញ្ចប់	31

សេចក្តីផ្តើម

នៅក្នុងជីវភាពរស់នៅប្រចាំថ្ងៃ ព័ត៌មានគឺជាផ្នែកមួយដ៏សំខាន់ដែលបានផ្តល់ជាចំនេះដឹង និងព្រឹត្តិការណ៍សំខាន់ៗដែលបានកើតឡើងនៅក្នុងសង្គមបច្ចុប្បន្ន។ បច្ចុប្បន្ននេះយើងសង្កេតឃើញថា ចំនួនអ្នកប្រើប្រាស់ Internet នៅកម្ពុជាមានការកើនឡើងយ៉ាងរហ័ស ហើយតម្រូវការនៃការចែករំលែកព័ត៌មាន និងចំនេះដឹងតាម Internet នេះក៏កាន់តែច្រើនឡើង ដូចនេះដើម្បីផ្តល់ភាពងាយស្រួលដល់អ្នកប្រើប្រាស់ អ្នកផ្តល់ព័ត៌មានឬចំណេះដឹងជាច្រើនអ្នកបានព្យាយាមរកនូវវិធីដើម្បីផ្តល់នូវព័ត៌មាន និងចំណេះដឹងទាំងអស់នោះទៅកាន់អ្នកប្រើប្រាស់អោយបានរហ័ស និងផ្តល់ភាពងាយស្រួលទៅកាន់អ្នកប្រើប្រាស់ផងដែរ។ ជម្រើសដែលសាកសមបំផុតនោះ គឺផ្តល់អត្ថបទទាំងនោះតាម គេហទំព័រ (Website)។ ក៏ប៉ុន្តែដើម្បីអាចផ្តល់អត្ថបទទាំងនោះតាមគេហទំព័របាន គឺតម្រូវអោយពួកគាត់ចេះបង្កើតនូវគេហទំព័រដោយខ្លួនឯង ឬក៏ជួលអ្នកជំនាញដើម្បីបង្កើតនូវគេហទំព័រអោយគាត់ ការប្រើប្រាស់មធ្យោបាយនេះតម្រូវអោយគាត់ចំណាយថវិកា និងពេលវេលាច្រើន ជាហេតុធ្វើអោយគាត់មានការលំបាកក្នុងការចែករំលែកនូវព័ត៌មានឬចំនេះដឹងទៅកាន់អ្នកប្រើប្រាស់បាន។

ដោយមើលឃើញនូវផលលំបាកដែល អ្នកផ្តល់ព័ត៌មាន និងអត្ថបទទាំងនោះ ទើបពួកយើងបានបង្កើតនូវប្រព័ន្ធ CMS មួយសំរាប់ជា Tool ក្នុងការបង្កើតគេហទំព័រដោយស្វ័យប្រវត្តិទៅអោយពួកគាត់ដើម្បីជាចំណែកមួយក្នុងការសម្រួលដល់ការចែករំលែកអត្ថបទឬបស់ពួកគាត់ទៅកាន់អ្នកប្រើប្រាស់។

ប្រជុំអត្ថបទ CMS មិនត្រឹមតែជា Tool សម្រាប់បង្កើតគេហទំព័រដល់អ្នក ប៉ុណ្ណោះទេ វាក៏ជាផ្នែកមួយដែលបានជួសសម្រួលសល់ការរស់នៅរបស់ប្រជាពលរដ្ឋខ្មែរ ដោយធ្វើអោយពួកគេមានប្រភពចែកចាយអត្ថបទបានកាន់តែច្រើន កាលណាប្រភពចែកចាយអត្ថបទកាន់តែច្រើន ការទទួលព័ត៌មាន ចំនេះដឹង ការកំសាន្តក៏កាន់តែកើនឡើងដូចគ្នា។

ពួកយើងសង្ឃឹមយ៉ាងមុតមាំថា ប្រព័ន្ធ CMS មួយនេះ នឹងជាជំនួយមួយដ៏សំខាន់សំរាប់ប្រជាពលរដ្ឋកម្ពុជា សំរាប់ការចែករំលែកនូវអត្ថបទទៅកាន់មនុស្សជាច្រើនទៀត តាមរយៈគេហទំព័រដែលបានបង្កើតឡើងដោយប្រើប្រាស់ប្រព័ន្ធ CMS មួយនេះ។

ប្រជុំអត្ថបទ CMS

១. គោលបំណង

ប្រជុំអត្ថបទ CMS ជាប្រភេទ Content Management System ដែលគ្រប់គ្រងទៅលើ ការគ្រប់គ្រងអត្ថបទ(Article Management) ដែលត្រូវបានបង្កើតឡើងក្នុងគោលបំណងដើម្បីជួយសម្រួលដល់អ្នកចង់ចែកចាយអត្ថបទលើ Internet ដោយប្រើប្រាស់គេហទំព័រផ្ទាល់ខ្លួន ប៉ុន្តែមិនមានជំនាញក្នុងការបង្កើតគេហទំព័រខ្លួនឯង និងមិនចង់ចំណាយថវិការដ្ឋបាលបង្កើតគេហទំព័រទាំងនោះ។ ប្រជុំអត្ថបទ CMS អាចចាត់ទុកជា Tool មួយ សំរាប់បង្កើតគេហទំព័រមួយដោយខ្លួនឯង មិនចាំបាច់ប្រើប្រាស់ជំនាញ IT និងមិនត្រូវការចំណាយថវិកាសំរាប់វាឡើយ។

២. រចនាសម្ព័ន្ធនៃការបង្កើតប្រព័ន្ធ

២.១ ជំហាននៃប្រព័ន្ធ

ប្រព័ន្ធ CMS នេះអាចធ្វើការគ្រប់គ្រងទៅលើ ៖

-User Management: ដែលគ្រប់គ្រងទៅលើ User ក្នុងនោះមានការបង្កើត User ថ្មី, ការកែប្រែ User Information និងការលុប User ចេញពីប្រព័ន្ធ។ ក្នុងនោះមាន User Role ចំនួន ៣ គឺ៖

- Admin : ជាអ្នកគ្រប់គ្រងទាំងមូលលើគេហទំព័រ
- Editor: ជាអ្នកចាត់ចែងលើអត្ថបទ
- Subscriber: ជាអ្នកប្រើប្រាស់គេហទំព័រទូទៅ

-Page Management: ដែលគ្រប់គ្រងទៅលើការបង្កើត Page ថ្មី, ការកែប្រែទៅលើ Page ចាស់ និងលុប Page ដែលមិនត្រូវការចេញ

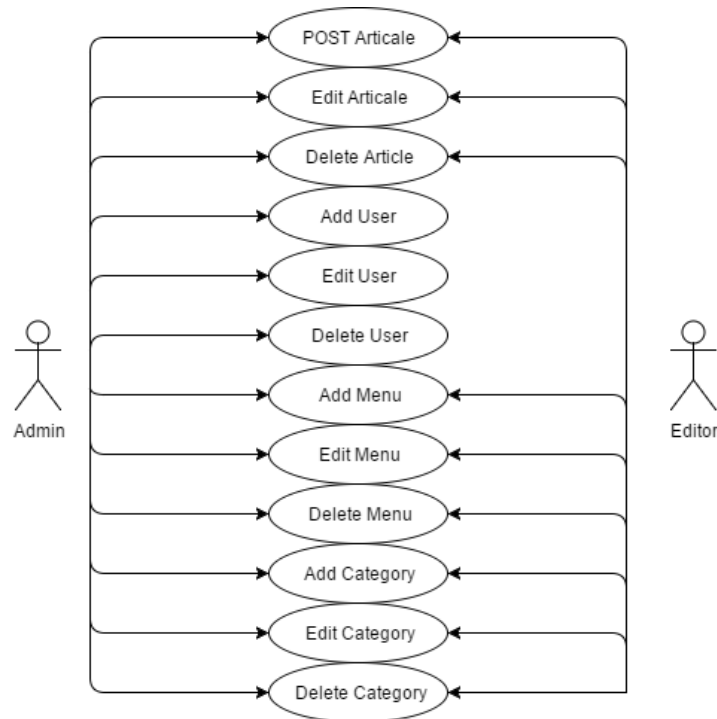
-Post Management: ដែលគ្រប់គ្រងលើការចែកចាយអត្ថបទនៅលើគេហទំព័រ ដែលរួមមានការបញ្ចូលអត្ថបទថ្មី, ការកែប្រែ និងលុបអត្ថបទនោះចោល។

-Menu Management: ដែលគ្រប់គ្រងលើ Menu នៅក្នុងគេហទំព័រ, អ្នកអាចធ្វើការបង្កើតនូវ menu ដោយខ្លួនឯង និងអាចកែប្រែវា ហើយលុបវាចោលបានផងដែរ។

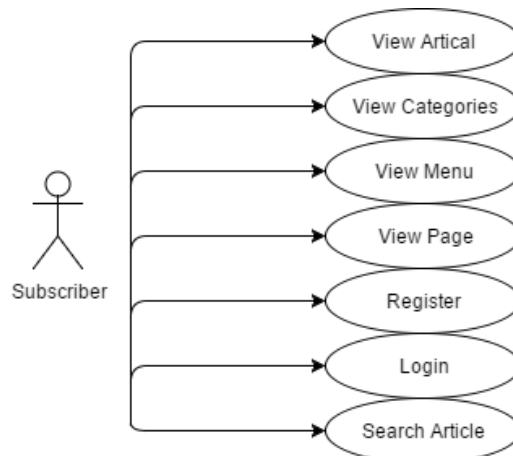
២.២ Use Case Diagram

នៅក្នុង ប្រជុំអត្ថបទ CMS មាន User Role ចំនួន ៣ គឺ Admin Editor និង Subscriber។ ដែល Use case Diagram ខាងក្រោមនឹងបង្ហាញពីសិទ្ធិ និងមុខងារដែល Role នីមួយៗអាចធ្វើបាន។

- Admin and Editor:



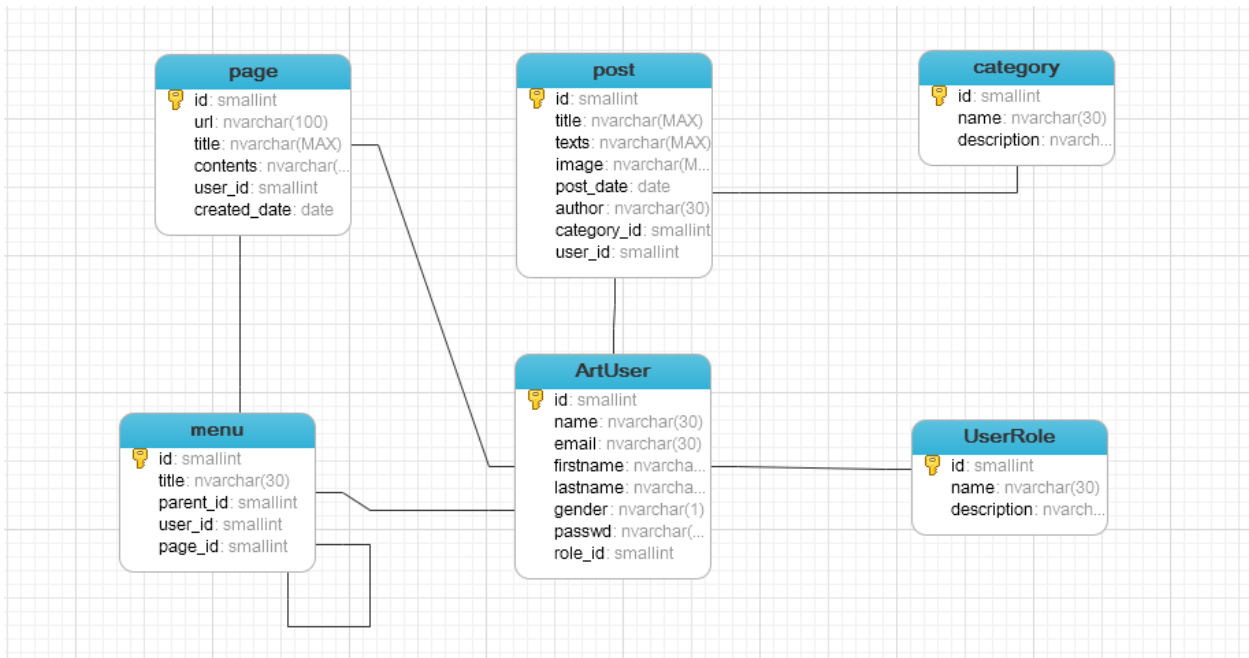
- Subscriber:



២.៣ Database Design

ប្រជុំអត្ថបទ CMS Database បានបង្កើតឡើងដោយ table ចំនួន 6 ដូចខាងក្រោម៖

- Page : សំរាប់ផ្ទុកទិន្នន័យរបស់ Page
- Post : សំរាប់ផ្ទុកទិន្នន័យរបស់ Post
- Category: សំរាប់ផ្ទុកទិន្នន័យរបស់ Category
- Menu: សំរាប់ផ្ទុកទិន្នន័យរបស់ Menu
- ArtUser: សំរាប់ផ្ទុកទិន្នន័យរបស់ User
- UserRole: សំរាប់ផ្ទុកទិន្នន័យរបស់ ប្រភេទ User Role



២.៤ Interface Design

២.៤.១ Back-End Design

ចំពោះខាង Back-End មានតែ User ដែលមាន role ជា Admin តែប៉ុណ្ណោះដែលអាចប្រើប្រាស់បាន។

- **Log-in Page** : User ត្រូវធ្វើការ Login សិនដើម្បីប្រើប្រាស់ប្រព័ន្ធ CMS នេះ។



- **User Management:** ការគ្រប់គ្រងលើ User
- A. Get All User

The screenshot shows the 'ប្រព័ន្ធអគ្គបទ' (CMS) interface. On the left is a sidebar with navigation links: គ្រប់គ្រងព័ត៌មាន (Manage Information), គ្រប់គ្រងទំព័រ (Manage Pages), គ្រប់គ្រងមីនុយ (Manage Menu), គ្រប់គ្រងអ្នកប្រើប្រាស់ (Manage Users), បន្ថែមអ្នកប្រើប្រាស់ (Add User), and អំពីយើង (About Us). The 'គ្រប់គ្រងអ្នកប្រើប្រាស់' link is selected. The main content area is titled 'អ្នកប្រើប្រាស់ទាំងអស់' (All Users). It features a table with columns: ឈ្មោះ-ឈ្មោះ (Name-Name), ឈ្មោះ (Name), អ៊ីម៉ែល (Email), មុខងារ (Position), and សកម្មភាព (Action). The table lists four users: rakky (admin), dara (editor), sethy (subscriber), and nara (editor). Each user has 'លុប' (Delete) and 'កែប្រែ' (Edit) buttons. A 'បង្កើតថ្មី' (Create New) button is at the top right. A footer note reads: រក្សាសិទ្ធិដោយប្រព័ន្ធព័ត៌មាន © ២០១៧.

ឈ្មោះ-ឈ្មោះ	ឈ្មោះ	អ៊ីម៉ែល	មុខងារ	សកម្មភាព
rakky	rakky sem	sem@gmail.com	admin	លុប កែប្រែ
dara	dara chem	dara@gmail.com	editor	លុប កែប្រែ
sethy	sethy proem	sethy@gmail.com	subscriber	លុប កែប្រែ
nara	nara nara	nara@gmail.com	editor	លុប កែប្រែ

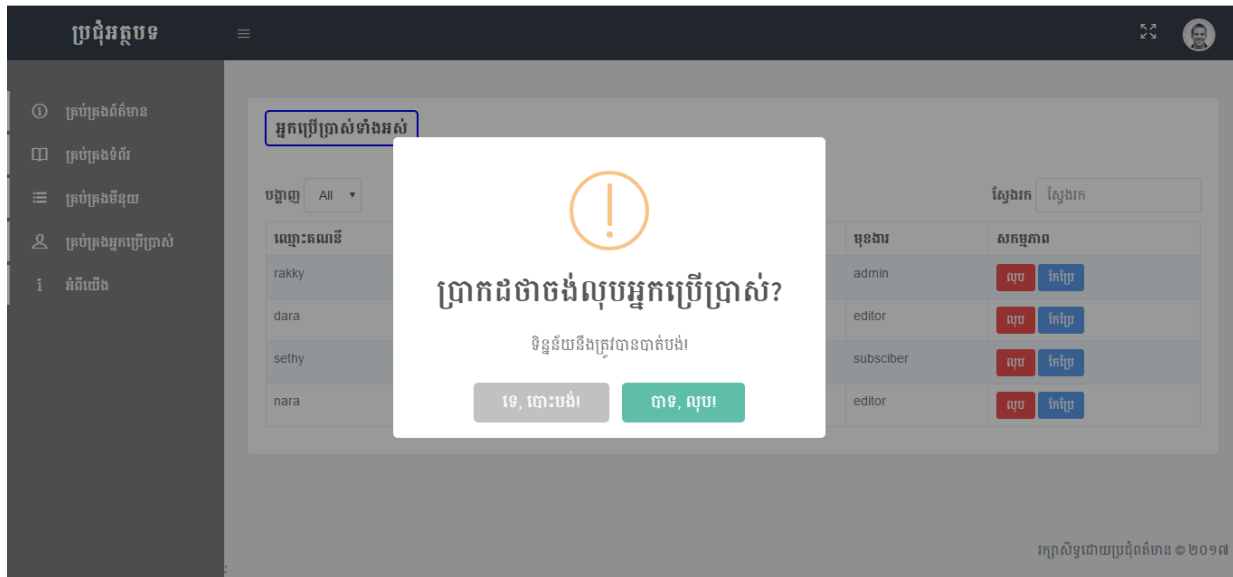
- B. User Profile

The screenshot shows the 'អំពីយើង' (About Us) page in the CMS. The sidebar is the same as the previous screenshot. The main content area is titled 'អំពីយើង' and contains a form for user profile information. The form fields are: ឈ្មោះ-ឈ្មោះ (Name-Name) with value 'rakky', អ៊ីម៉ែល (Email) with value 'sem@gmail.com', នាម (Name) with value 'rakky', នាមត្រកូល (Surname) with value 'sem', ភេទ (Gender) with a dropdown menu showing 'ប្រុស' (Male), and តំណភ្ជាប់ (Link) with value 'admin'. A 'កែប្រែ' (Edit) button is at the bottom right.

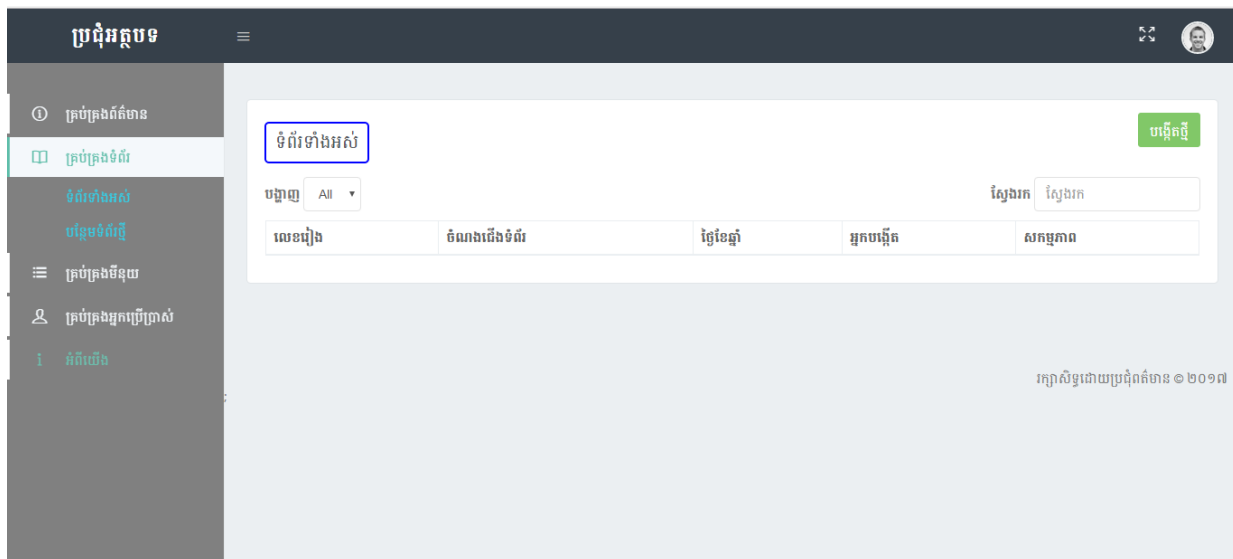
- C. Add New User

- D. Update User

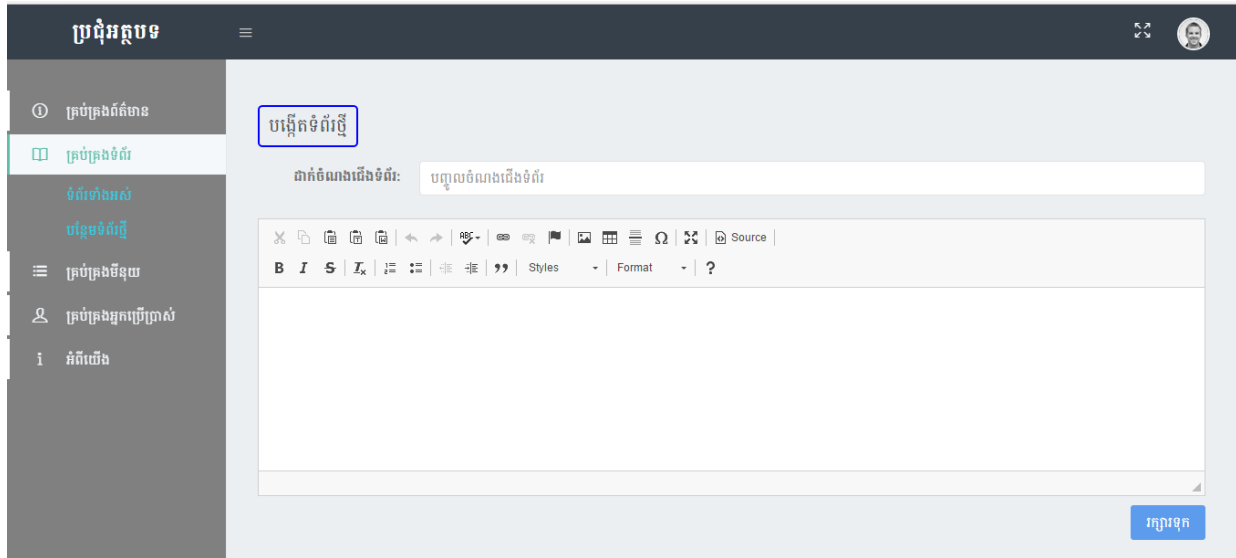
- E. Delete User



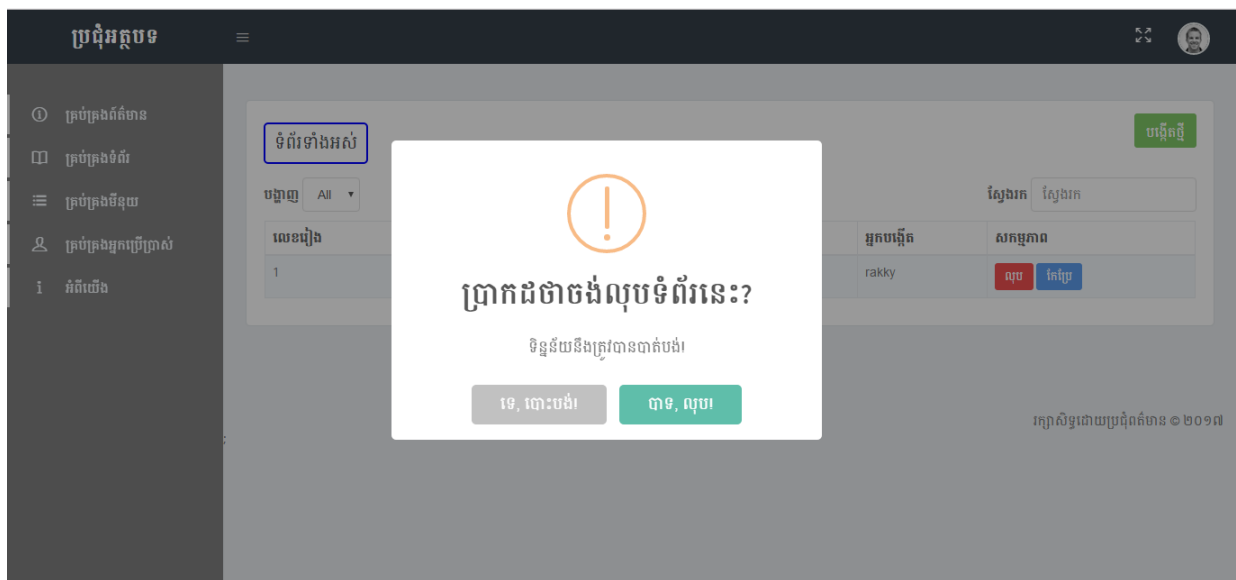
- **Page Management** : ការគ្រប់គ្រងទៅលើ Page របស់គេហទំព័រ
- A. Get All Pages



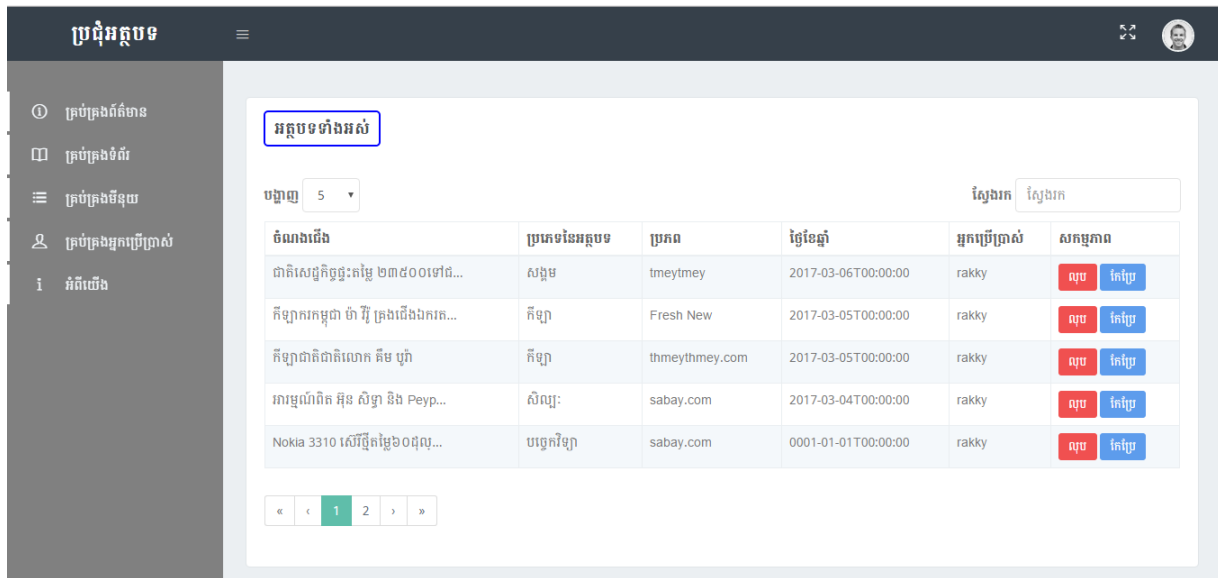
- B. Create New Page



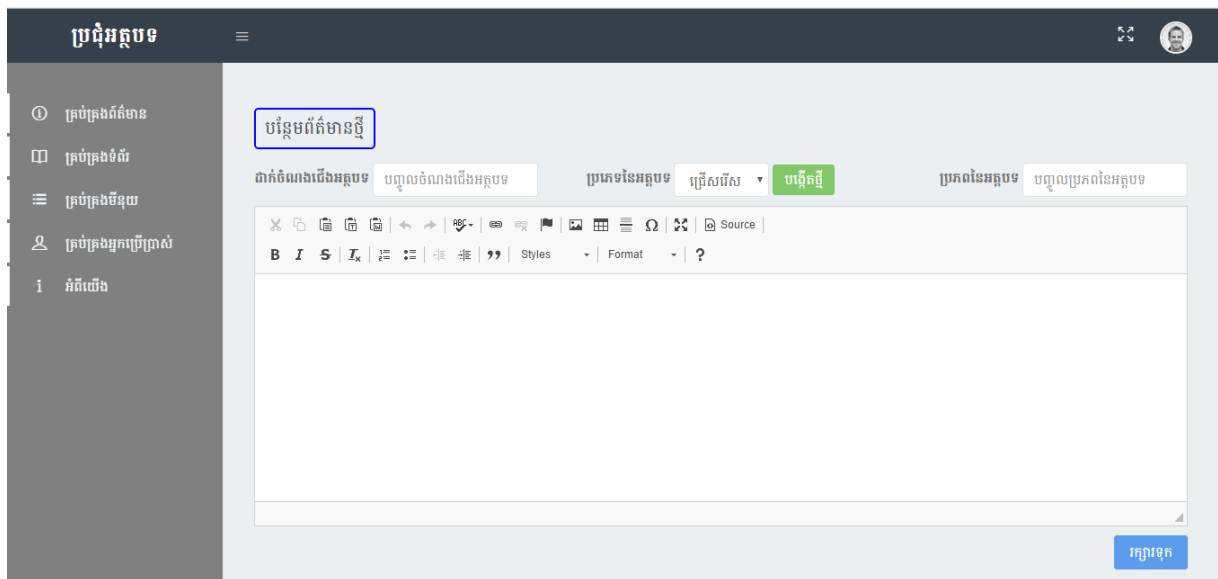
- C. Delete Page



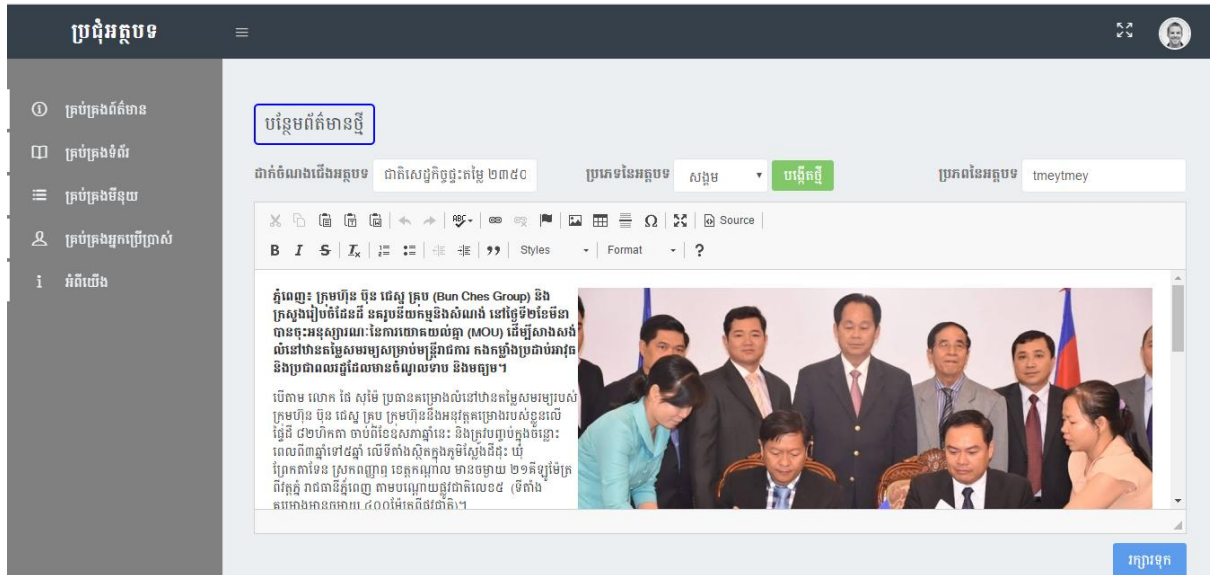
- **Post Management:** ការគ្រប់គ្រងលើអត្ថបទ។
- A. All Posts



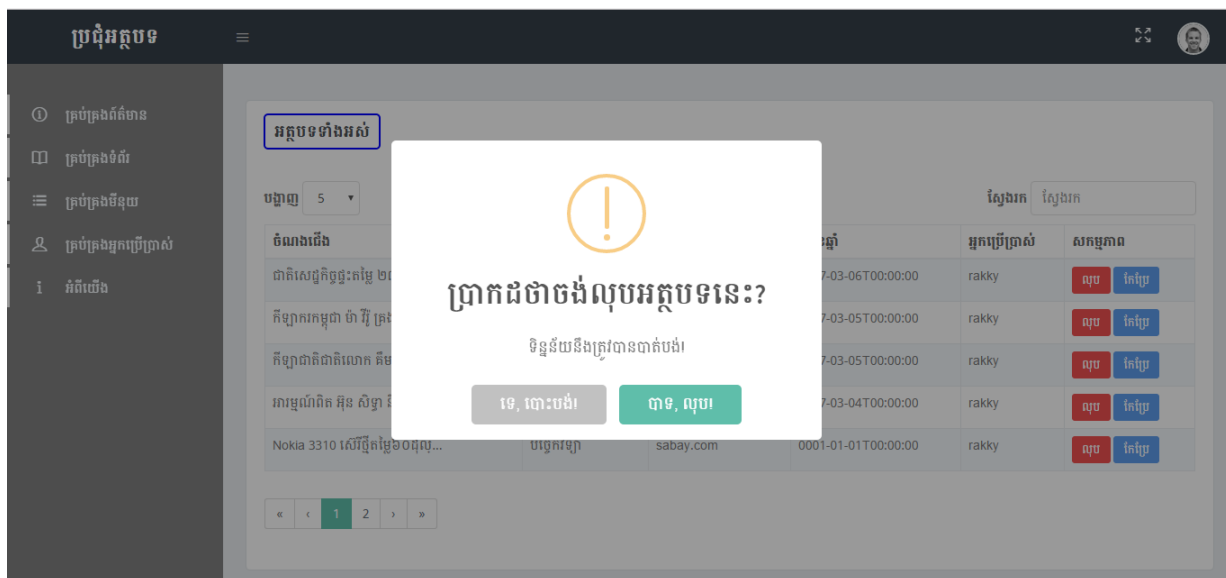
- B. Add New Post



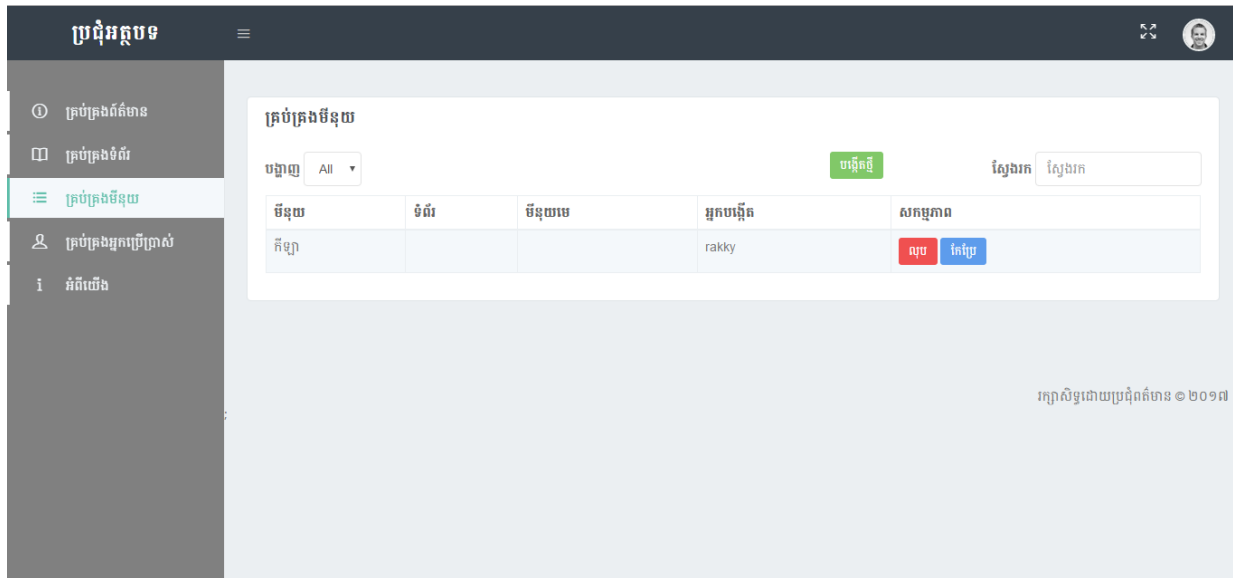
- C- Edit Post



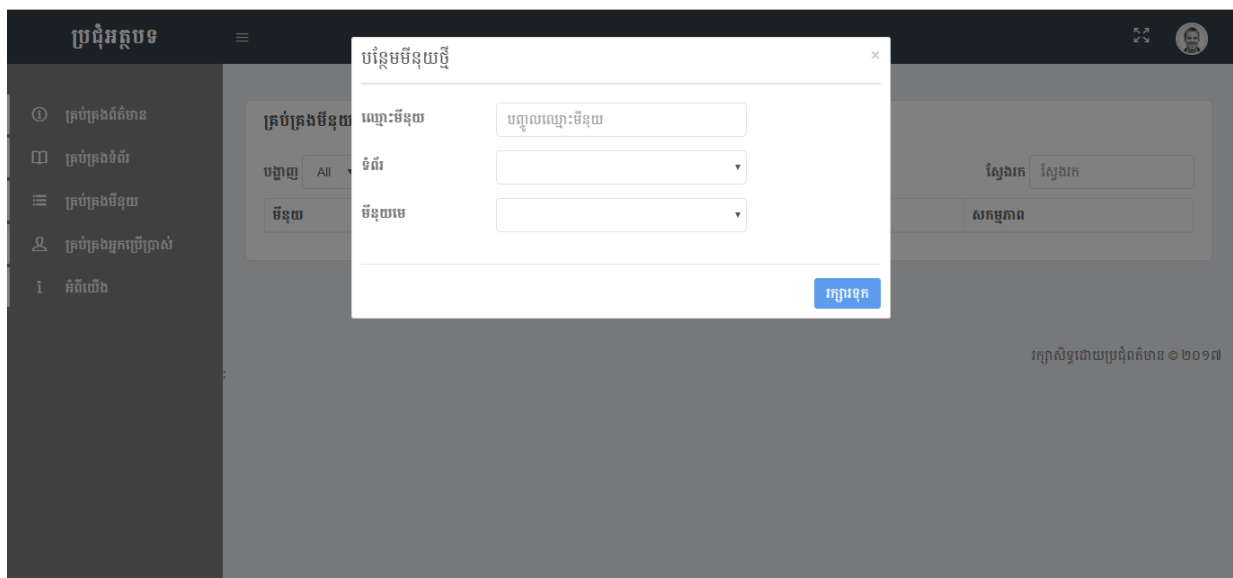
- D. Delete Post



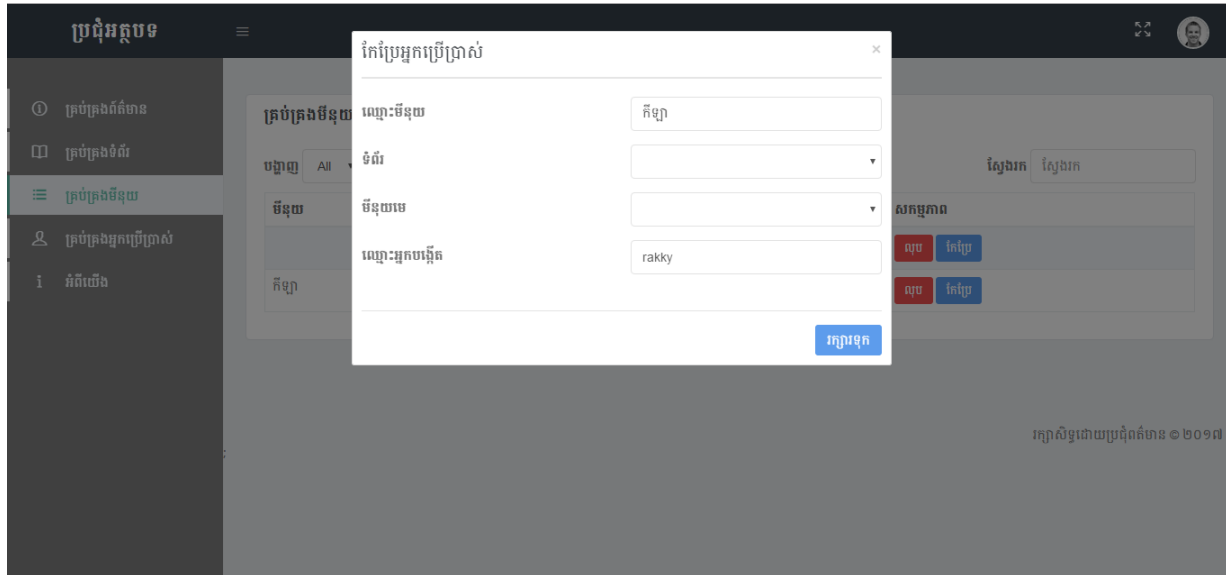
- **Menu Management:** កាត្រប់គ្រងទៅលើ Menu
- A. Get All Menu



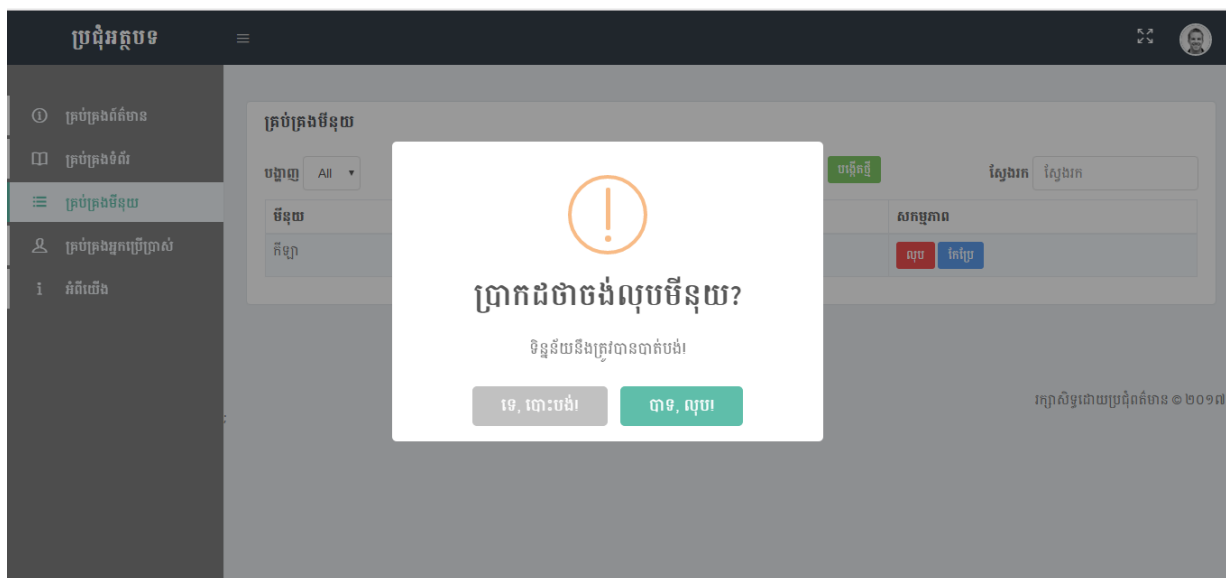
- B. Create New Menu



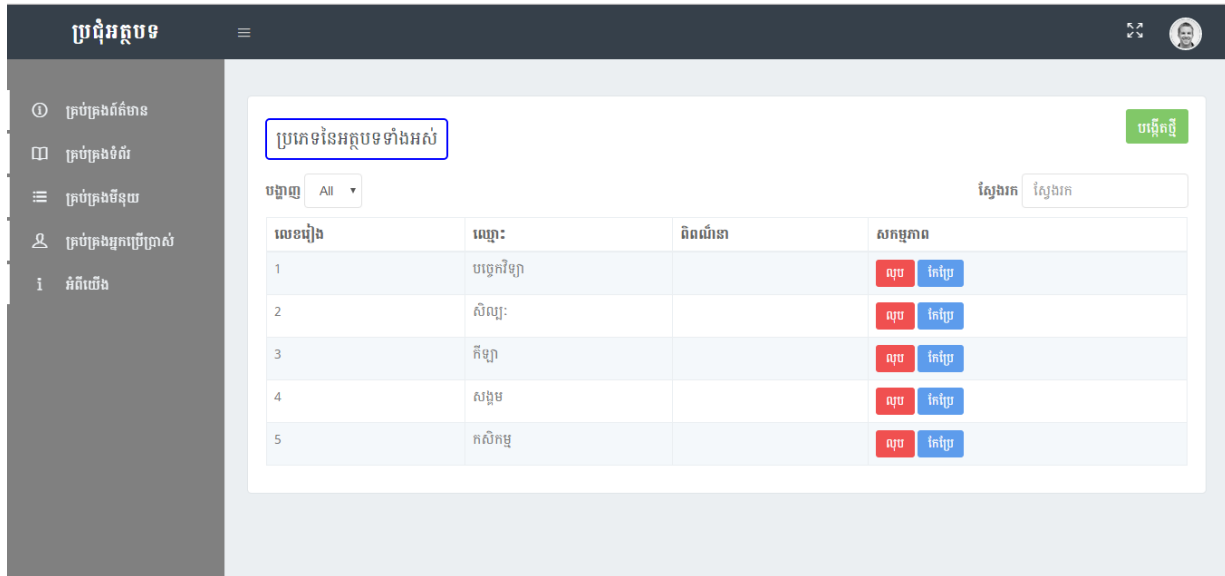
- C. Edit Menu



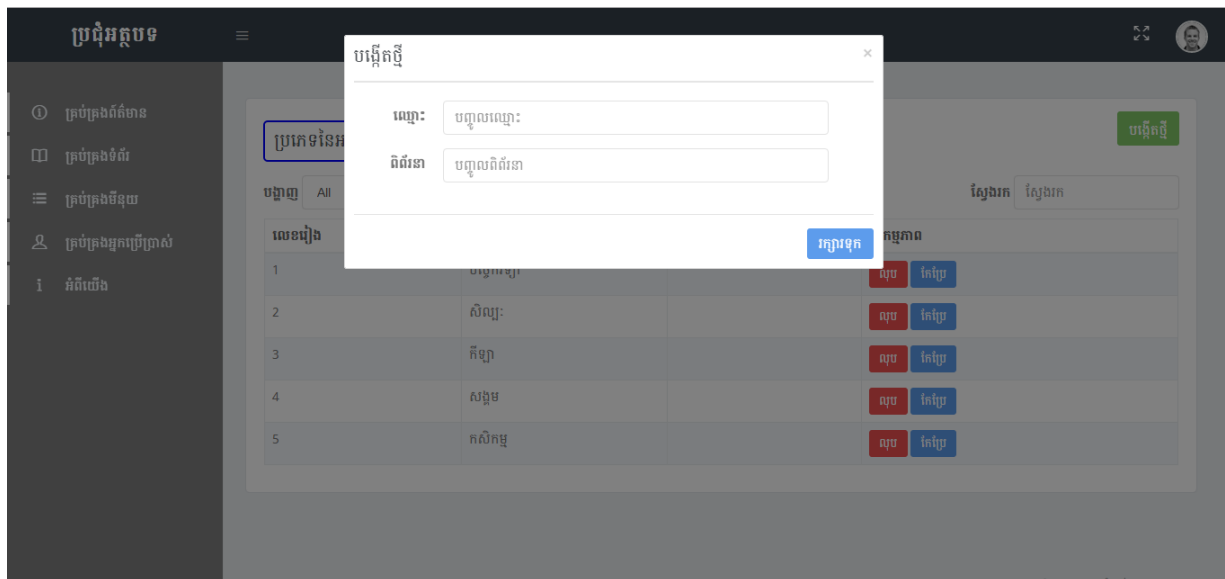
- D. Delete Menu



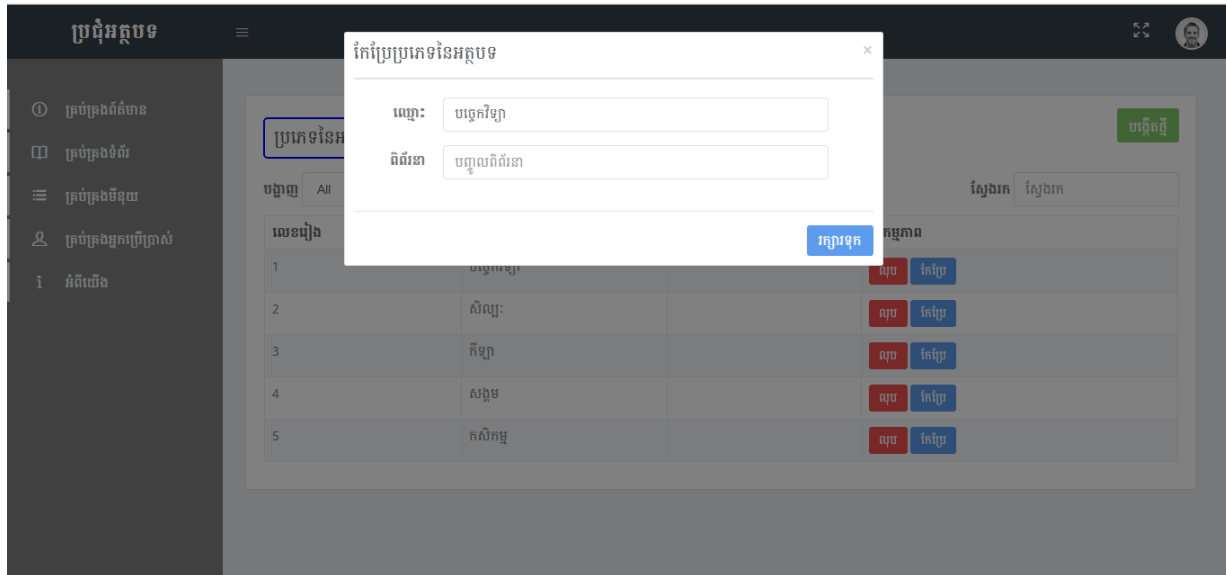
- **Category Management:** ការគ្រប់គ្រងលើប្រភេទ Categories
- A. All Categories



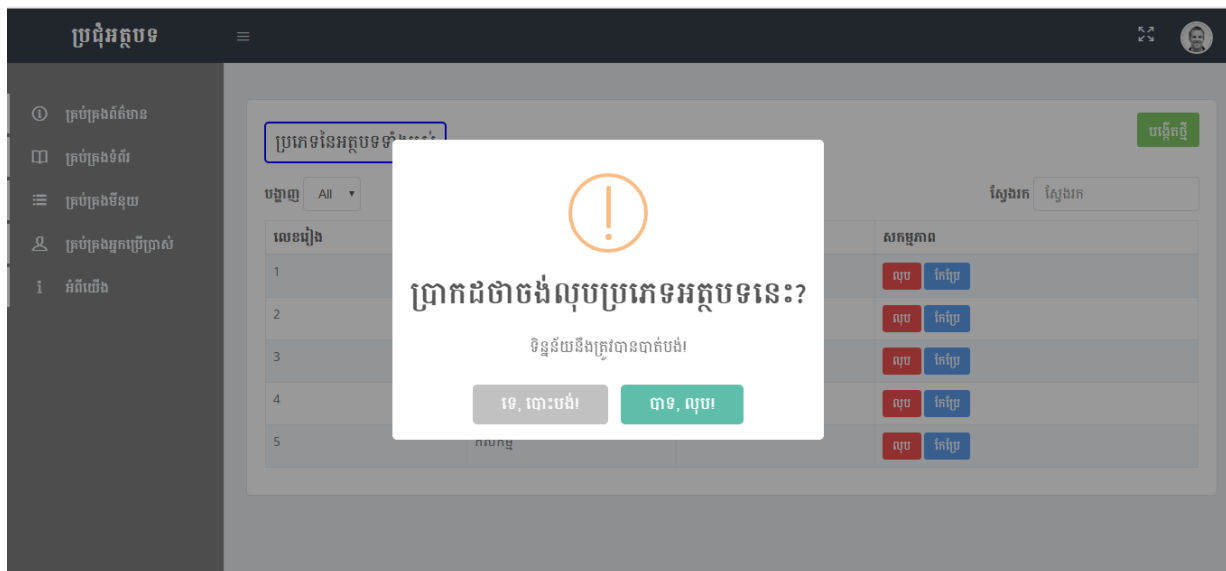
- B. Add New Category



- C. Edit Category

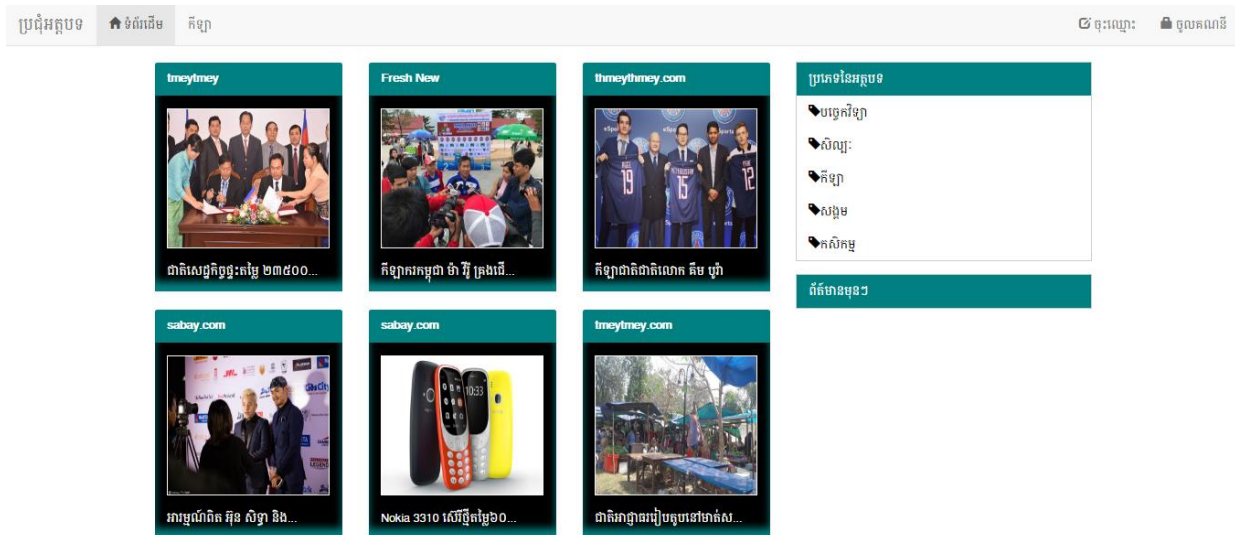


- D. Delete category



២.៤.២ Front-End Design

- A. Home Page



- B. Page Detail




C. Detail Post

ប្រព័ន្ធអគ្គបទ
🏠 ទំព័រដើម
📄 កីឡា
➡ គណនី

Nokia 3310 សេរីថ្មីតម្លៃ៦០ដុល្លារ

👤 rakky
📅 បច្ចេកវិទ្យា
🕒 0001-01-01T00:00:00



នៅឆ្នាំ២០១៧ ក្រុមហ៊ុនទូរសព្ទយក្សពីអតីតកាលមួយនេះបាន ចេញទូរសព្ទ ៣៣១០ ម្តងទៀតជាមួយការរចនាថ្មីតាមបែបទំនើប និងមានការផលិត ចំនួន១០០លានគ្រឿង ហើយតម្លៃវិញត្រឹម៦០ដុល្លារប៉ុណ្ណោះ។ យ៉ាងណាក៏ដោយ មានក្រុមហ៊ុនមួយបានយកទូរសព្ទ៣៣១០ ទៅរចនាបន្ថែមដ៏គួរ ឲ្យគ្រាក់ឆ្កើល គឺស្រោបមានប្រមុខប្រធានាធិបតីស៊ី Puma និងដាក់ក្នុងប្រអប់ ក្របាច់មានបិទបិទមិត្តសញ្ញាប្រទេសស្រី ដែលការកែច្នៃនេះធ្វើឲ្យ ៣៣១០ តម្លៃ៦០ ដុល្លារ ត្រូវលក់រហូតដល់ ១៧០០ដុល្លារក្រោយទឹក។

ប្រភេទនៃអត្ថបទ

- 👁 បច្ចេកវិទ្យា
- 👁 សិល្បៈ
- 👁 កីឡា
- 👁 សង្គម
- 👁 កសិកម្ម

ព័ត៌មានមុនៗ


រក្សាសិទ្ធិដោយសមាជិកក្រុម

D. Category Page


ប្រព័ន្ធអគ្គបទ
🏠 ទំព័រដើម
📄 កីឡា
➡ គណនី



ជាតិអាជ្ញាធររៀបចំនៅមាត់ស...



ជាតិសេដ្ឋកិច្ចថ្មី៖តម្លៃ ២៣៥០០...



សាលាខ្មែរណា ឲ្យបន្តរៀន...

ប្រភេទនៃអត្ថបទ

- 👁 បច្ចេកវិទ្យា
- 👁 សិល្បៈ
- 👁 កីឡា
- 👁 សង្គម
- 👁 កសិកម្ម

ព័ត៌មានមុនៗ

រក្សាសិទ្ធិដោយសមាជិកក្រុម

២.៥ API Document

ប្រជុំអត្ថបទ CMS បង្កើតឡើងរួមជាមួយនឹង API ដែលងាយស្រួលសម្រាប់ការបង្កើត Website និង អាចប្រើបានចំពោះ Platform ផ្សេងៗទៀត។

Part	Method	URL	Request	Response	File
USER_ROLE	GET	api/UserRole/GetUserroleById/{id}		{ "\$id": "int", "id": int, "name": "string", "description": "string", "users": [] }	UserRoleController
	GET	api/UserRole/GetAllUserRole		[{ "id": int, "name": "string", "description": "string", "users": [] }]	
	POST	api/UserRole/PostUserrole	{ "id": int, "name": "string", "description": "string" }	{ "id": int, "name": "string", "description": "string" }	
	PUT	api/UserRole/PutUserrole	{ "id": int, "name": "string", "description": "string" }	{ "id": int, "name": "string", "description": "string" }	
	DELETE	api/UserRole/DeleteuserroleById/{id}	id : int	{ "data": "string" }	
USER	GET	/api/User/GetAllUsers		[{ "\$id": "int", "Id": int, "Name": "string", "Email": "string", "Firstname": "string", "Lastname": "string", "Gender": "string(1)", "Passwd": "string", "RoleName": "string", "RoleDescription": "string" }	UserController

				<pre> }] </pre>	
	GET	/api/User/GetAllWithRole		<pre> [{ "\$id": "int", "id": int, "name": "string", "email": "string", "firstname": "string", "lastname": "string", "gender": "string(1)", "passwd": "string", "role_id": int, "role": { "\$id": "int", "id": int, "name": "string", "description": "string", "users": [] }, "menus": [], "pages": [], "posts": [] }] </pre>	
	GET	/api/User/GetUserById/{id}	Id : int	<pre> [{ "\$id": "int", "Id": int, "Name": "string", "Email": "string", "Firstname": "string", "Lastname": "string", "Gender": "string(1)", "Passwd": "string", "RoleName": "string", "RoleDescription": "string" }] </pre>	
	GET	/api/User/GetSession?name={name}&password={password}	name : "string" password : "string"	<pre> { "\$id": "int", "Id": int, "Name": "string", "Email": "string", "Firstname": "string", "Lastname": "string", "Gender": "string(1)", "Passwd": null, "RoleName": "string", "RoleDescription": null } </pre>	
	POST	/api/User/PostUser	<pre> { "id": int, </pre>	<pre> { "id": int, "name": "string", </pre>	

			<pre>"name": "string", "email": "string", "firstname": "string", "lastname": "string", "gender": "string", "passwd": "string", "role_id": int }</pre>	<pre>"email": "string", "firstname": "string", "lastname": "string", "gender": "string", "passwd": "string", "role_id": int }</pre>	
	PUT	/api/User/PutUser	<pre>{ "id": int, "name": "string", "email": "string", "firstname": "string", "lastname": "string", "gender": "string", "passwd": "string", "role_id": int }</pre>	<pre>{ "id": int, "name": "string", "email": "string", "firstname": "string", "lastname": "string", "gender": "string", "passwd": "string", "role_id": int }</pre>	
	DELETE	/api/User/DeleteUserById/{Id}	Id : int	<pre>{ "id": int, "name": "string", "email": "string", "firstname": "string", "lastname": "string", "gender": "string", "passwd": "string", "role_id": int }</pre>	
POST	GET	/api/Post/GetAllPosts		<pre>[{ "\$id": "int", "id": int, "title": "string", "texts": "string", "image": "string", "post_date": "date", "author": "string", "cat_id": int, "category_name": "string", "username": "string", "status": boolean }]</pre>	PostController
	GET	/api/Post/Get/{id}	Id : int	<pre>{ "\$id": "int", "id": int,</pre>	

				<pre> "title": "string", "texts": "string", "image": "string", "post_date": "date", "author": "string", "cat_id": int, "category_name": "string", "username": "string", "status": boolean } </pre>	
	POST		<pre> { "id": int, "title": "string", "texts": "string", "image": "string", "post_date": "date", "author": "string", "category_id": int, "user_id": int } </pre>		
	UPDATE	/api/Post/Put	<pre> { "id":int, "title":"st ring", "texts":"st ring", "image":"st ring", "post_date" :"date", "author":"s tring", "category_i d":int, "user_id":i nt, "user":{" }, "category": { } } </pre>	<pre> { "id":int, "title":"string", "texts":"string", "image":"string", "post_date":"date", "author":"string", "category_id":int, "user_id":int, "user":{" }, "category":{" } } </pre>	
	DELETE	api/Post/Delete/{id}	id : int	<pre> { "data" : "string" } </pre>	
PAGE	GET	/api/Page/Get		<pre> [{ "id": int, "url": "string", "title": "string", "contents": "string", "user_id": int, </pre>	PageCo ntroll er

				"created_date": "date" }]	
	GET	api/Page/Get/{Id}		{ "id": int, "url": "string", "title": "string", "contents": "string", "user_id": int, "created_date": "date" }	
	POST	/api/Page/Post	{ "id": int, "url": "string", "title": "string", "contents": "string", "user_id": int, "created_date" : "date" }	{ "id": int, "url": "string", "title": "string", "contents": "string", "user_id": int, "created_date": "date" }	
	PUT	/api/Page/Put	{ "id": int, "url": "string", "title": "string", "contents": "string", "user_id": int, "created_date" : "date" }	{ "id": int, "url": "string", "title": "string", "contents": "string", "user_id": int, "created_date": "date" }	
	DELETE	/api/Page/Delete/{id}	int : int	{ "data" : "string" }	
MENU	GET	api/Menu/Get		[{ "id": int, "title": "string", "parent_id": int, "page_id": int, "user": { }, "submenu": [{}], "parent": {} }]	MenuCo ntroll er
	POST	/api/Menu/Post	{ "id": int, "title": "string", "parent_id": int, }	{ "id": int, "title": "string", "parent_id": int, "user_id": int, "page_id": int }	

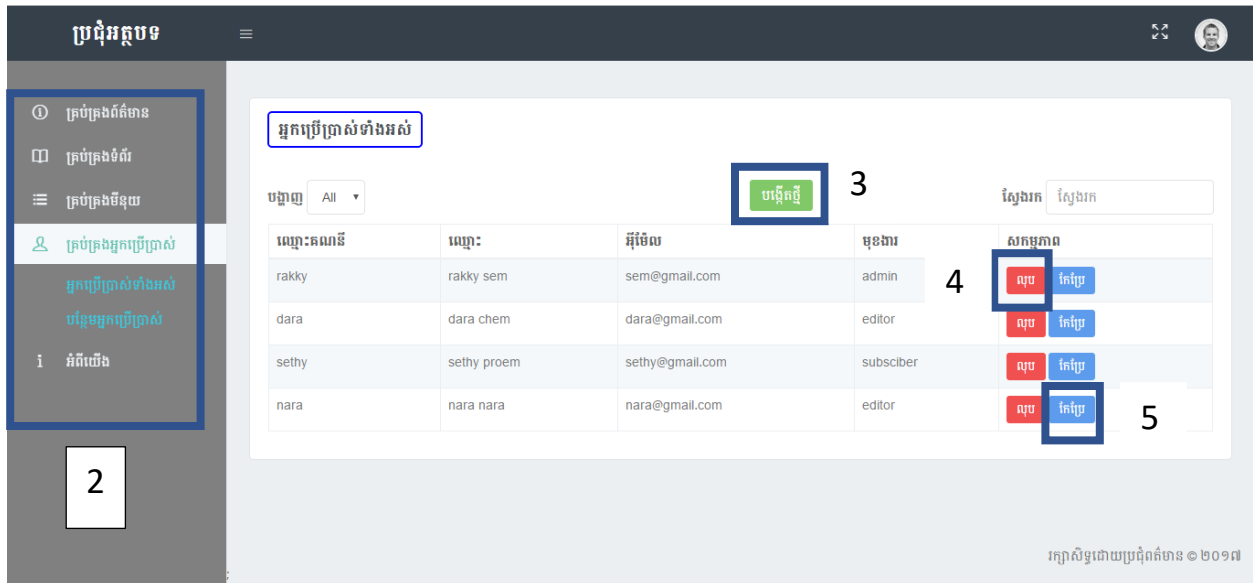
			"user_id": int, "page_id": int }	}	
	PUT	/api/Menu/Put	{ "id": int, "title": "string", "parent_id": int, "user_id": int, "page_id": int }	{ "id": int, "title": "string", "parent_id": int, "user_id": int, "page_id": int }	
	DELETE	api/Menu/Delete/{id}	id : int	{ "data": "string" }	
CATEGORY	GET	api/Category/GetAllCategories		{ "id": int, "name": "string", "description": "string", }	CategoryController
	POST		{ "id": int, "name": "string", "description": "string", }	{ "id": int, "name": "string", "description": "string", }	
	UPDATE	api/Category/Put	{ "id": int, "name": "string", "description": "string", }	{ "id": int, "name": "string", "description": "string", }	
	DELETE	api/Category/DeleteCategoryByID/{id}	id : int	{ "data" : "string" }	

៣. Implementation

៣.១ Application Flow

- Admin

១. មុនដំបូង Admin ត្រូវ Login ជាមុនសិន



២. បន្ទាប់មក គាត់អាចធ្វើការគ្រប់គ្រងទៅលើ User, Post, Menu,និង Page បាន

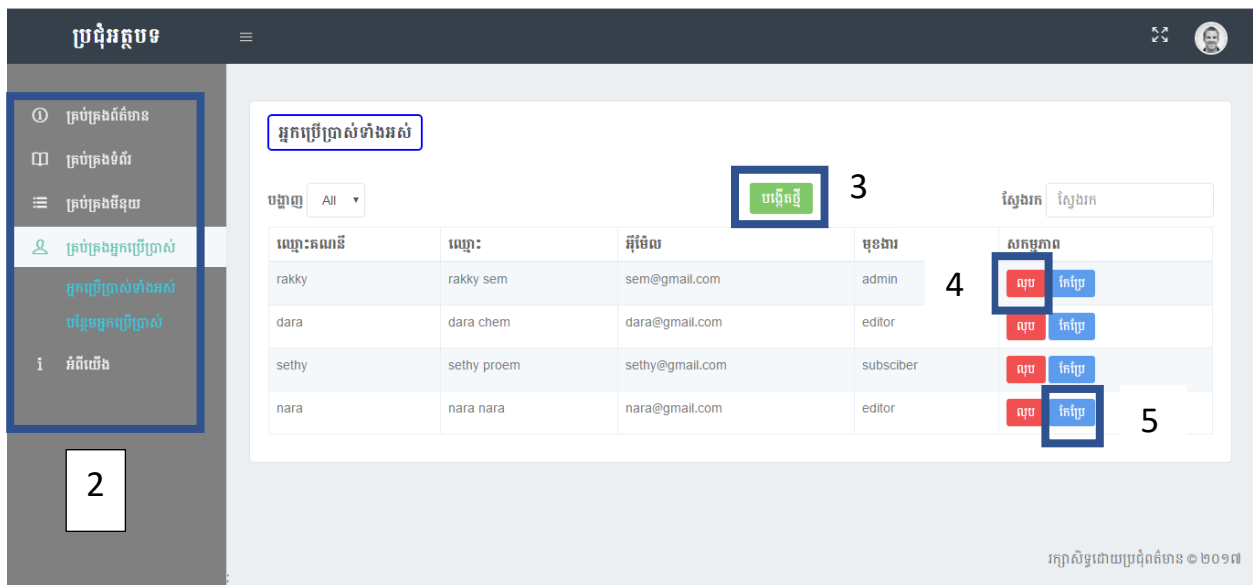
៣. ស្វែងរក ប៊ូតុង “បង្កើតថ្មី” ដើម្បីបង្កើត User, Post, Menu,និង Page ថ្មី

៤. ស្វែងរក ប៊ូតុង “លុប” ដើម្បីលុប User, Post, Menu,និង Page ចេញ

៥. ស្វែងរក ប៊ូតុង “កែប្រែ” ដើម្បីកែប្រែព័ត៌មានរបស់ User, Post, Menu,និង Page

- Editor

១. មុនដំបូង Editor ត្រូវ Login ជាមុនសិន



២. បន្ទាប់មក គាត់អាចធ្វើការគ្រប់គ្រងទៅលើ Post, Menu,និង Page បាន តែមិនអាចធ្វើការគ្រប់គ្រងទៅលើ User នោះទេ។

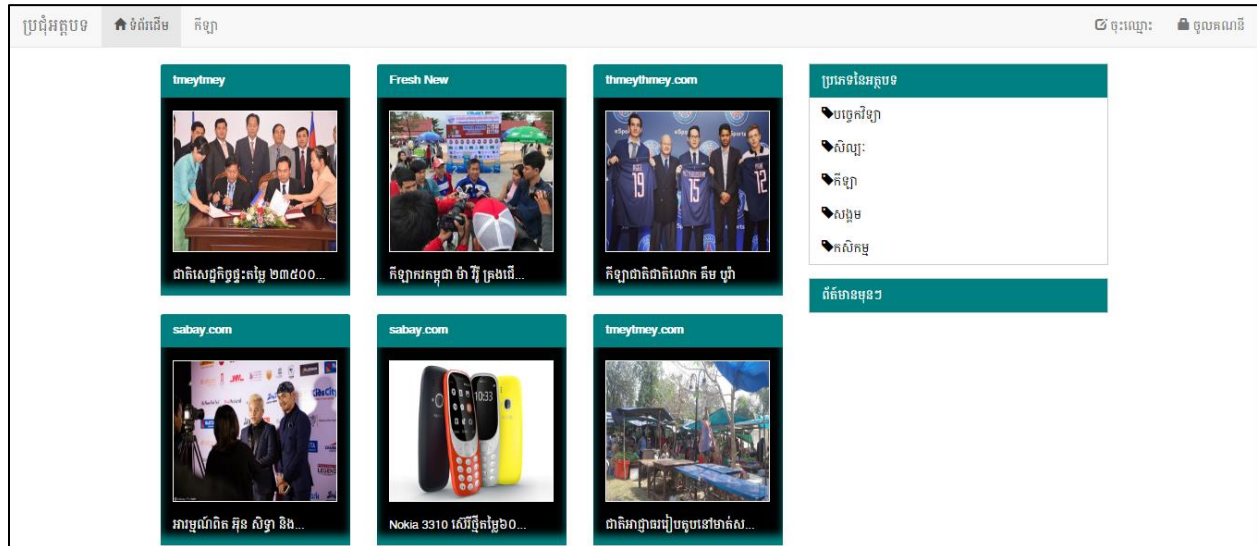
៣. ស្វែងរក ប៊ូតុង “បង្កើតថ្មី” ដើម្បីបង្កើត Post, Menu,និង Page ថ្មី

៤. ស្វែងរក ប៊ូតុង “លុប” ដើម្បីលុប Post, Menu,និង Page ចេញ

៥. ស្វែងរក ប៊ូតុង “កែប្រែ” ដើម្បីកែប្រែព័ត៌មានរបស់ Post, Menu,និង Page

- Subscriber

១. មុនដំបូង Subscriber ត្រូវ ចូលមកកាន់គេហទំព័រជាមុនសិន



២. បន្ទាប់មក គាត់អាចធ្វើការView ទៅលើ Article ដែលបាន Post នៅលើគេហទំព័រ

៣. គាត់អាចស្វែងរក Article ដែលគាត់ចង់អានបានតាម Menu, category និង Search

៤. គាត់អាច Register ទៅកាន់គេហទំព័របាន

៥. គាត់អាចធ្វើការ Login មកកាន់គេហទំព័របាន។

៣.២ បច្ចេកវិទ្យាប្រើប្រាស់

+Client Side

Html : សំរាប់រៀបចំជាគ្រោង រូបរាង គេហទំព័រ

Css : សំរាប់រចនាគេហទំព័រអោយមានភាពស្រស់ស្អាត

Javascript : សំរាប់ធ្វើអោយគេហទំព័រមានភាព Dynamic

Bootstrap : សំរាប់ធ្វើអោយគេហទំព័រ Responsive

Angular : សំរាប់រៀប ទិន្នន័យដែលបានមកពី Database មករៀបចំលើ គេហទំព័រ

+Server Side

Asp.Net MVC : សំរាប់បង្កើត Web Forms-based applications។

+Web Service

Asp.Net API : សំរាប់បង្កើតជា Web service API ដែលប្រើសំរាប់ជាផ្លូវនាំData ពី Client ទៅ Server

+Database

SQLServer 2008 R2

៤. ការពិភាក្សា

៤.១ ចំណុចខ្លាំង ចំណុចខ្សោយរបស់ប្រព័ន្ធ

បន្ទាប់ពីប្រព័ន្ធនេះ បានស្ថាបនាចរាចរណ៍រួចមកយើងសង្កេតឃើញថា ប្រព័ន្ធមានចំណុចខ្លាំងនិងចំណុចខ្សោយដូចជា៖

ក. ចំណុចខ្លាំង

- វាដំណើរការបានល្អ
- មុខងារទាំងអស់ដែលមាននៅក្នុងប្រព័ន្ធ ដំណើរការបានជោគជ័យ
- អ្នកប្រើប្រាស់អាចយកទៅប្រើប្រាស់បាន
- មិនមានភាព Error

ខ. ចំណុចខ្សោយ

- Interface មិនទាន់ទាក់ទាញ ស្រស់ស្អាត
- មានចំណុចជាច្រើនទៀតដែលត្រូវបន្ថែម
- ការបង្កើត Page មិនទាន់មាន layout ច្រើន អោយអ្នកប្រើប្រាស់

៤.២ គំរោងអនាគត

ប៉ុន្តែពួកយើងនៅតែបន្ត ធ្វើគំរោងសំរាប់ពេលអនាគតសំរាប់ប្រព័ន្ធមួយនេះ៖

- បន្ថែមមុខងារដែលត្រូវការបន្ថែម
- រៀបចំបន្ថែមនូវ Layout អោយកាន់តែមានភាពសំបូររំបែប
- ការរៀបចំធ្វើ Savelist, Downloading, Read Later...និងច្រើនទៀត
- ការរៀបចំប្រព័ន្ធអោយដំណើរការលើ ទូរស័ព្ទដៃទំនើប Smartphone, Tablet។

៥. សេចក្តីបញ្ចប់

បន្ទាប់ពីឆ្លងកាត់ការបង្កើត CMS នេះអស់មួយខែរួមមក យើងសំរេចបាននូវ CMS មួយដែលមានសមត្ថភាពអាចបង្កើតនូវគេហទំព័រមួយសមរម្យជូនទៅដល់ អ្នកដែលមានគោលបំណងចង់ចែកចាយអត្ថបទដោយប្រើប្រាស់នូវគេហទំព័រលើInternet ។ បន្ទាប់ពីគាត់ប្រើប្រាស់ ប្រជុំអត្ថបទ CMS របស់យើងរួចមកគាត់នឹងអាចចែករំលែកនូវរាល់ព័ត៌មាន ចំនេះដឹង ការកំសាន្ត ទៅកាន់មនុស្សរាប់លានអ្នកផ្សេងទៀតដែលជាឱកាសក្នុងការស្វែងរកនូវចំនេះដឹង និងព័ត៌មាននៅក្នុងសង្គមរស់នៅបច្ចុប្បន្ន។

ពួកយើងសង្ឃឹមថាការសិក្សាស្រាវជ្រាវរបស់ពួកយើងប្រាកដជាមានអ្នកប្រើប្រាស់ច្រើន ហើយបានផ្តល់ប្រយោជន៍ជាច្រើនទៅកាន់សង្គមជាតិនាពេលបច្ចុប្បន្ន និងអនាគតផងដែរ។