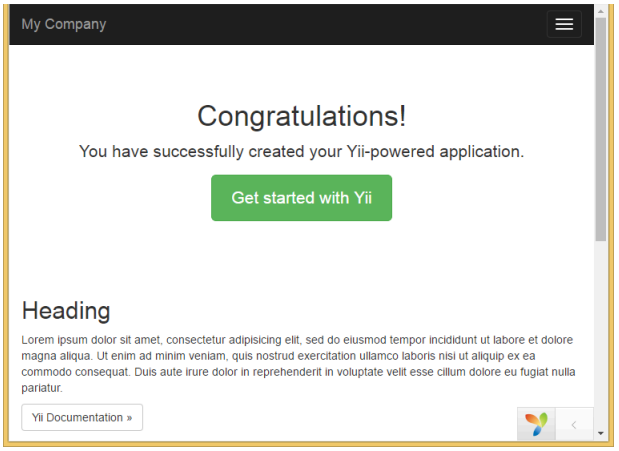
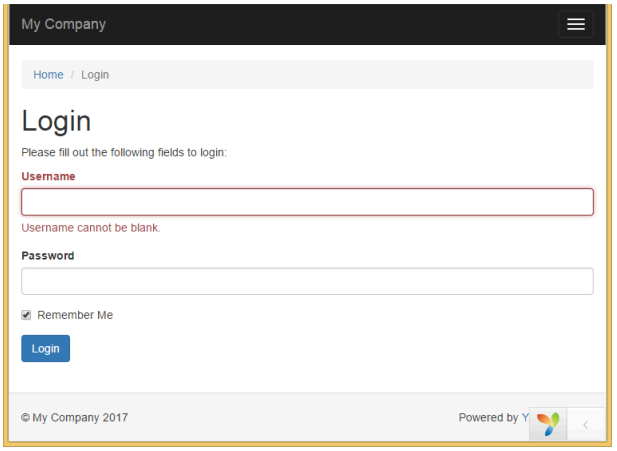
Лабораторна робота №2. Установка розширеного шаблону Yii2 (Advanced). Реєстрація користувачів. Генерація CRUD-коду (Gii).

Завдання 1. Встановлення шаблону проекту Advanced Template.

1. Встановіть та ініціюйте фреймворк Yii2 з шаблоном *Advanced* в проект *yii-advanced.* Доступ до веб-сайту та адмінки повинен здійснюватись за такими адресами:

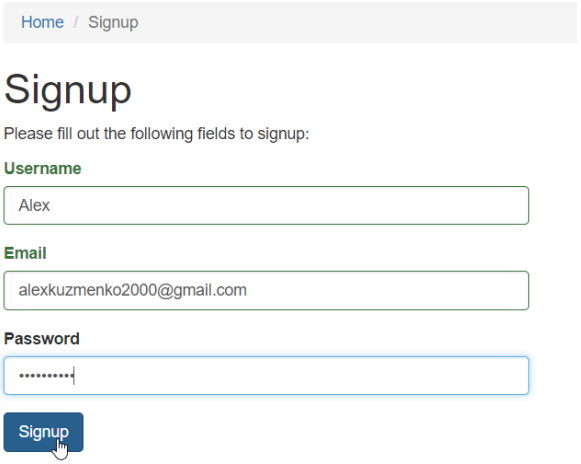
`http://yii-client.local/`	`http://yii-admin.local/`
	

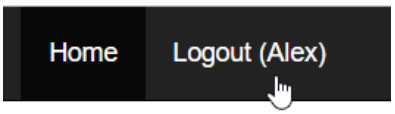
2. Додайте проект в середовище розробки
3. Виконайте ініціалізацію проекту (php init). Вкажіть режим розробки.
4. Створіть базу даних *yii_advanced*. Налаштуйте з'єднання до бази даних (common/config/main-local.php).
5. Зробіть міграцію бази. Для цього в командному рядку в корені проекту виконайте команду:

```
php yii migrate
```

В базі даних повинні з'явитись таблиці *user* та *migration*.

6. В користувацькій частині створіть обліковий запис. Протестуйте функції входу:



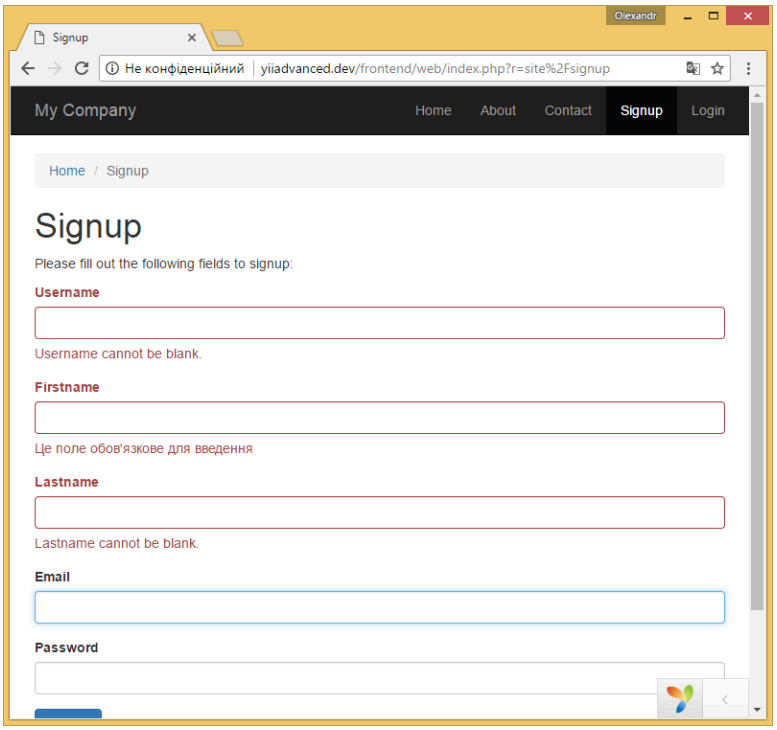


Завдання 2. Реєстрація користувача. Додавання та налаштування додаткових полів

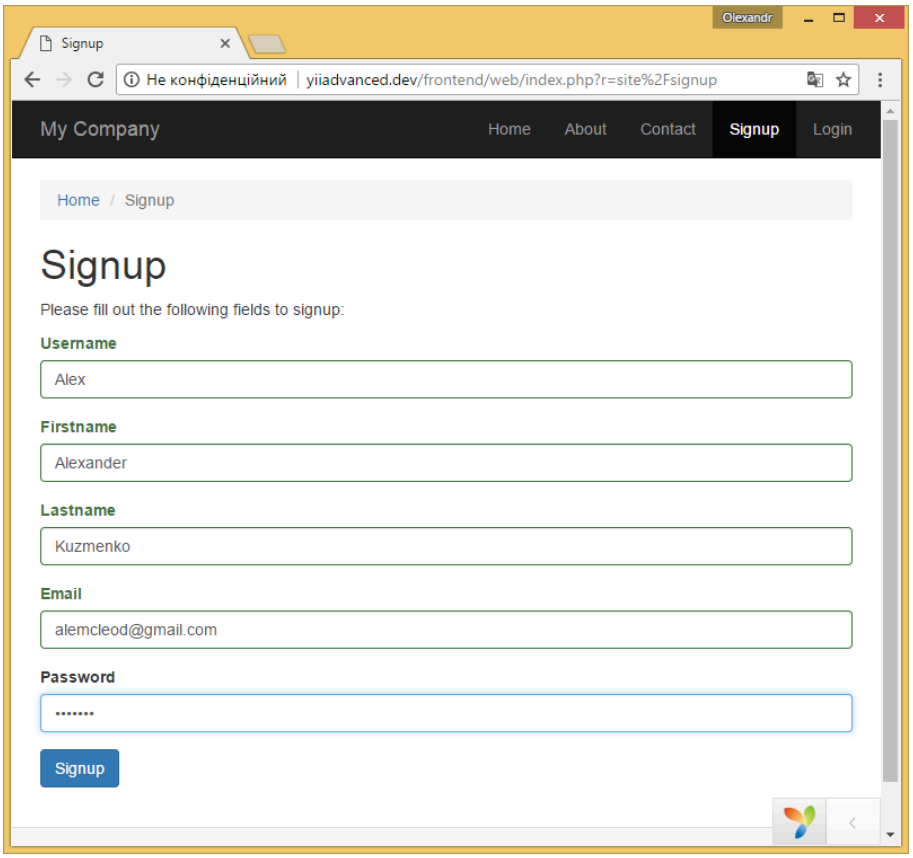
1. В таблиці *user* додайте поля firstname і lastname:

1	**id**	int(11)	
2	**username**	varchar(255)	utf8_unicode_ci
3	**firstname**	varchar(255)	utf8_unicode_ci
4	**lastname**	varchar(255)	utf8_unicode_ci
5	**auth_key**	varchar(32)	utf8_unicode_ci
6	**password_hash**	varchar(255)	utf8_unicode_ci
7	**password_reset_token**	varchar(255)	utf8_unicode_ci
8	**email**	varchar(255)	utf8_unicode_ci
9	**status**	smallint(6)	
10	**created_at**	int(11)	
11	**updated_at**	int(11)	

2. В клас моделі, що відповідає за реєстрацію користувачів, додайте *доступні* властивості *firstname* і *lastname*.
3. В тому ж класі в метод `rules()` додайте правила для валідації полів *firstname* і *lastname*: вони повинні бути обов'язковими для введення.
4. В правило валідації для поля *firstname* додайте елемент з ключем *message* та рядковим значенням «Це поле обов'язкове для введення»
5. Перевірте роботу форми:



6. В метод `signup()` моделі реєстрації користувача додайте ініціалізацію відповідних властивостей об'єкта `$user`.
7. Перевірте роботу форми реєстрації:



Завдання 3. Генерація коду з допомогою інструменту Gii

1. Знайомство з Gii: https://yiiframework.com.ua/ru/doc/guide/2/start-gii/
2. Створіть таблицю post з такими полями:

id	int(11)	
title	varchar(255)	utf8_general_ci
description	text	utf8_general_ci

3. Перейдіть в інструмент `gii`:

`http://{{hostname}}/index.php?r=gii`

4. Створимо модель класу `ActiveRecord` (для роботи з базою даних). Перейдіть в Генератор моделі. Виберіть таблицю `post`, та задайте ім'я моделі `Post`
5. Згенеруйте модель. Переконайтесь, що файл з класом моделі *Post.php* з'явився в папці *models*.
6. Згенеруємо CRUD код. Перейдіть в CRUD Generator і задайте такі імена класів моделей і контролера:

Клас моделі *frontend\models\Post*

Клас моделі для пошуку та сортування *frontend\models\PostSearch*

Клас контролера задайте *frontend\controllers\PostController*

Після генерації CRUD-кода преконайтесь в наявності таких файлів в проекті:

controllers/PostController.php

models/Post.php

models/PostSearch.php

views/post/.php*

7. Перейдіть по URL: `http://{{hostname}}/index.php?r=post`
8. Використаємо фільтр контролю доступу *AccessControl* для роботи з публікаціями лише авторизованим користувачам. Для цього в класі контролеру *PostController* модифікуйте метод `behaviors()`:

Перевірте тепер, чи є можливість додавати публікації неавторизованим користувачам.