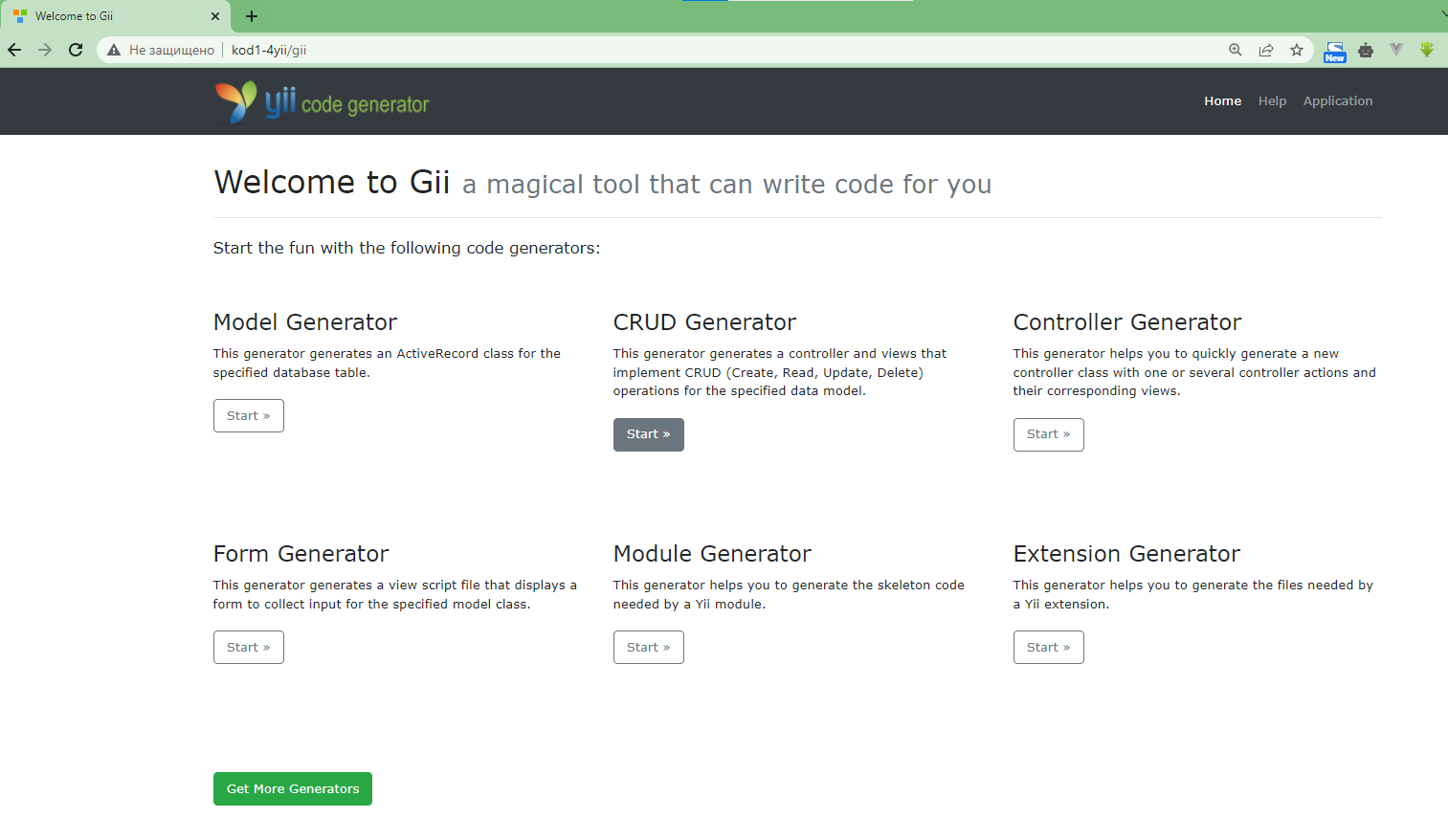
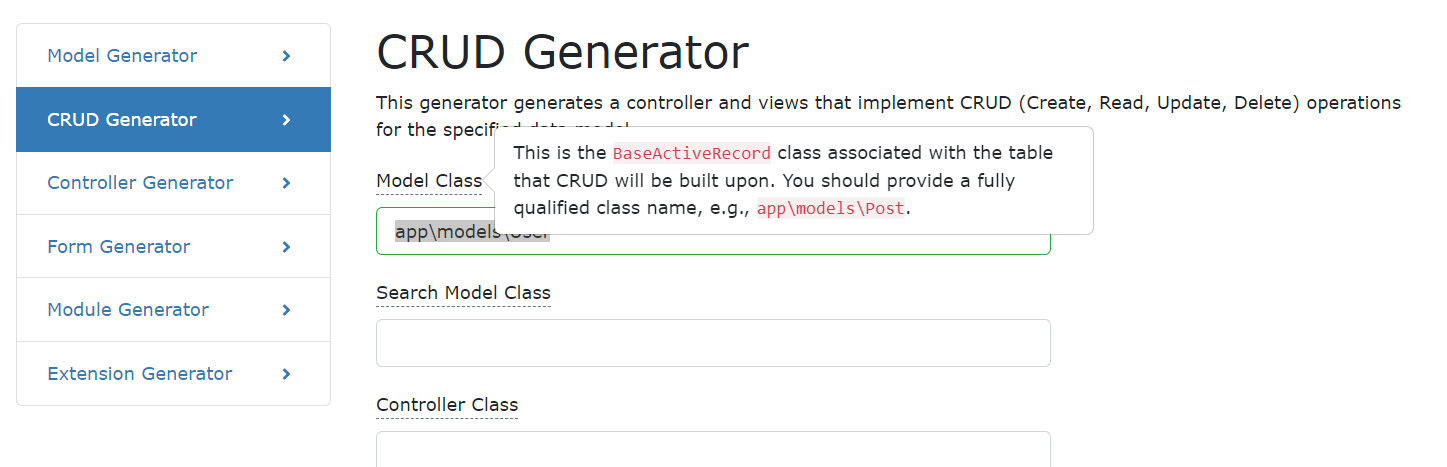
**Регистрация пользователя**

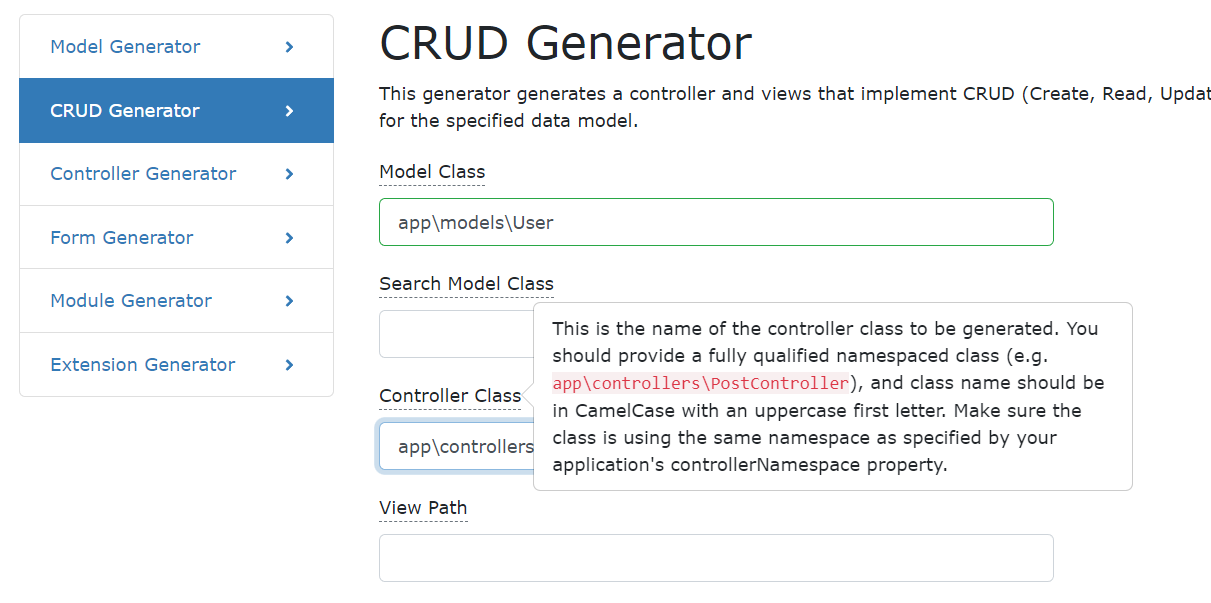
1. В браузере заходим в генератор кода **Gii** и выбираем **CRUD Generator**



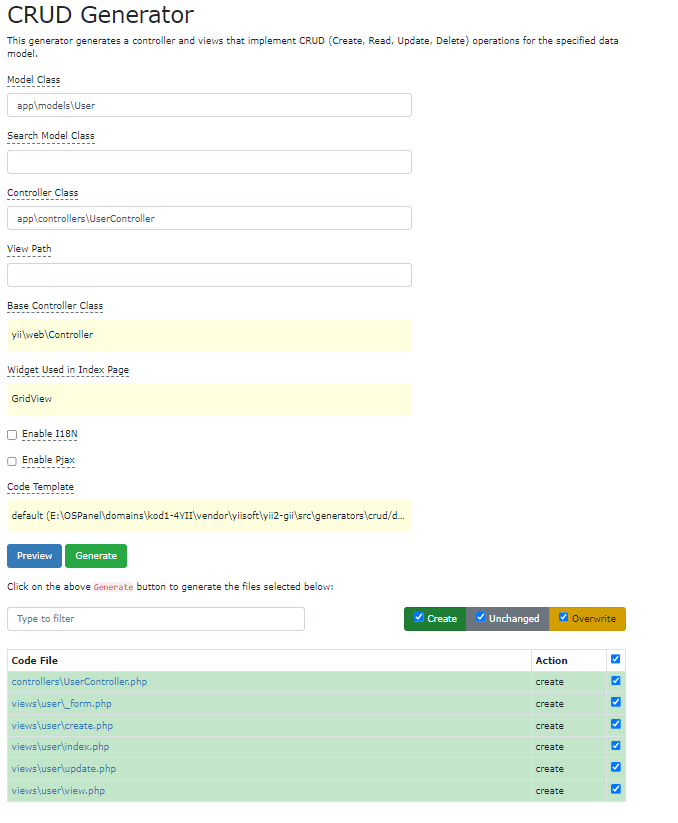
1. Навести мышку на **Model Class** на всплывающей подсказке красным цветом выделена подсказка app\models\Post какой код надо ввести в это поле. Только мы создаем класс **User - app\models\User**



1. Также заполняем поле **Controller Class** app\controllers\UserController



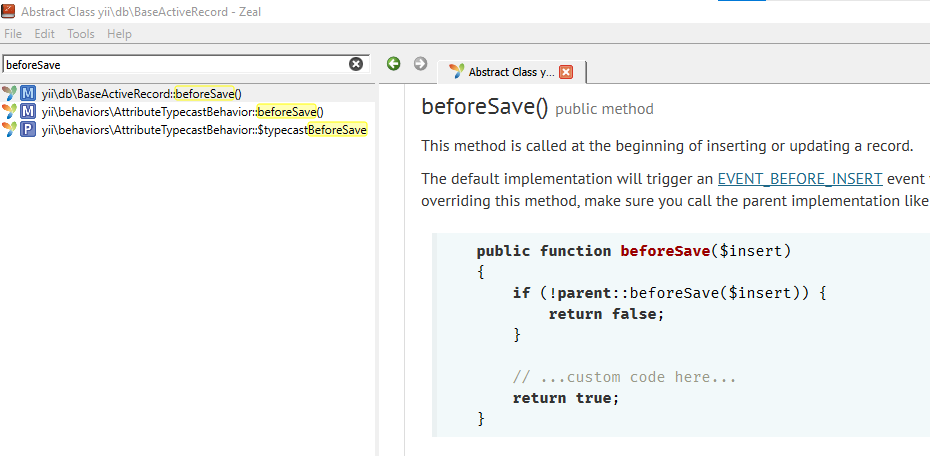
1. Нажать кнопку Preview. Внизу видим таблицу создаваемых контроллеров и представлений.



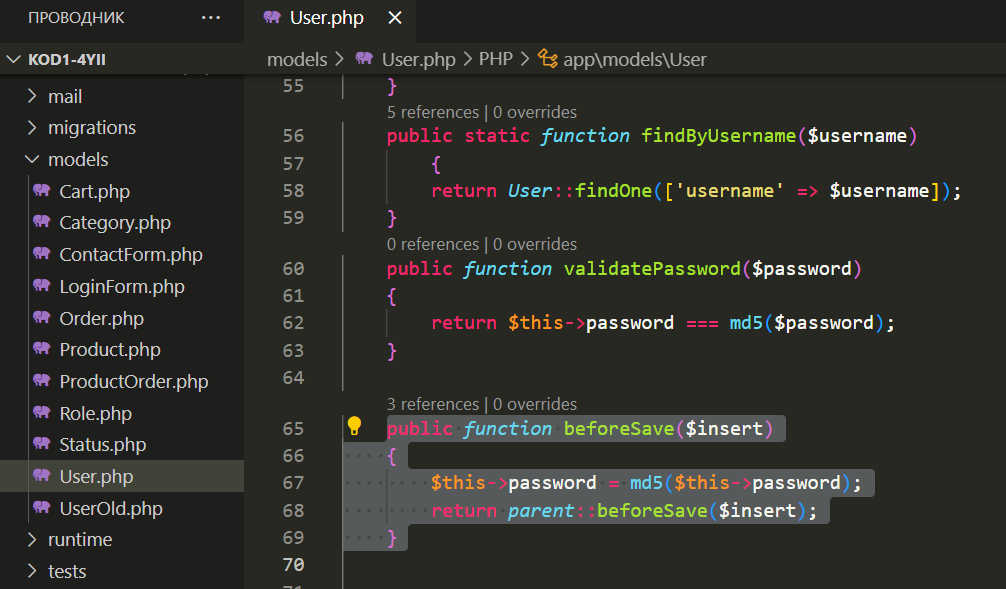
1. Нажать кнопку Generate. Внизу откроется окно с подтверждением создания.



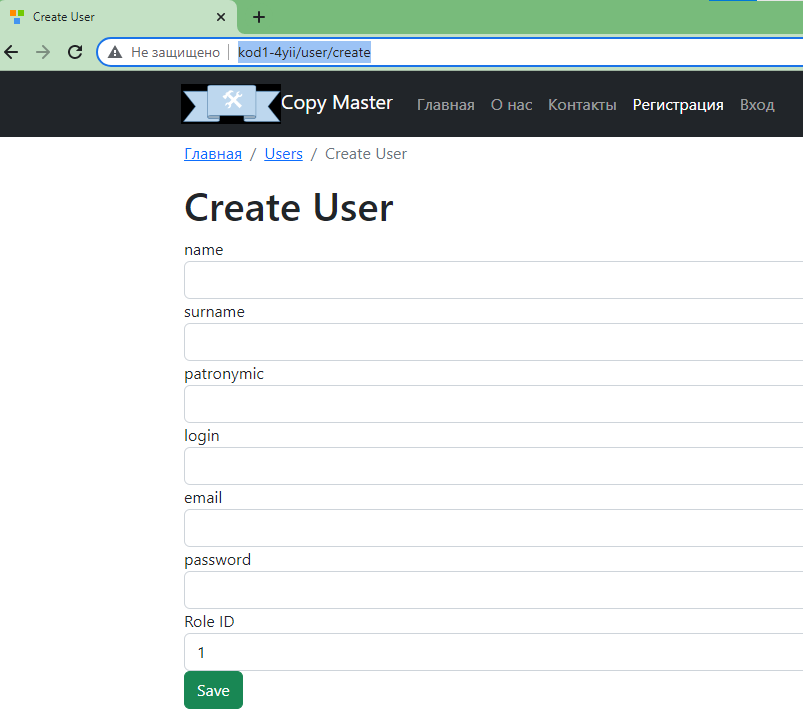
1. Для шифрования пароля при добавлении в базу данных используется метод **beforeSave**. В **ZEAL** в строке поиска ввести **beforeSave** и в первой же статье скопировать этот метод.



1. Вставить в конец класса **User** в **models\User.php** и исправить, добавив функцию шифрования md5.

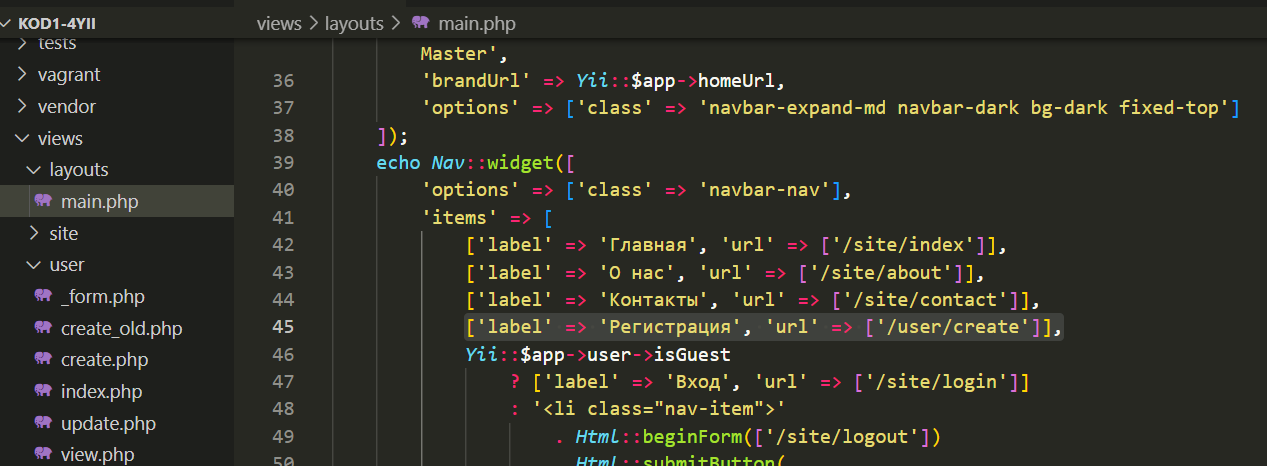


1. Проверить, что регистрация работает в командной строке браузера на главной странице проекта начать писать **user** автоматически допишется адрес страницы добавления пользователя.

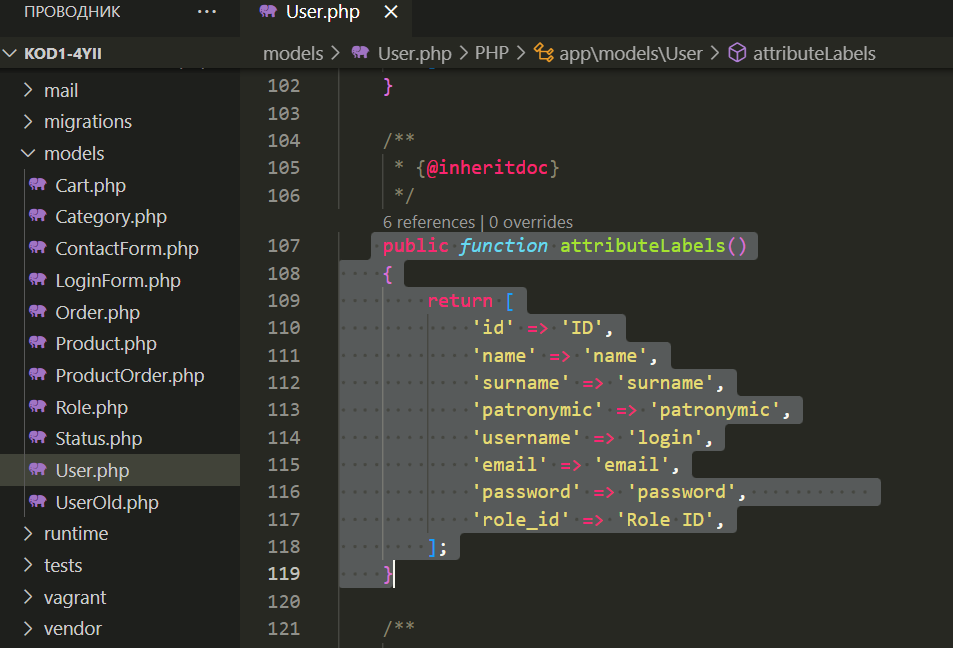


Запомнить этот путь.   
Заполнить таблицу данными и нажать кнопку **Save**, затем проверить в базе данных, что пользователь добавился.

1. Добавить кнопку Регистрация в меню проекта надо в файле **views\layouts\main.php** в секции **widget** добавить строку с кнопкой **['label' => 'Регистрация', 'url' => ['/user/create']],**



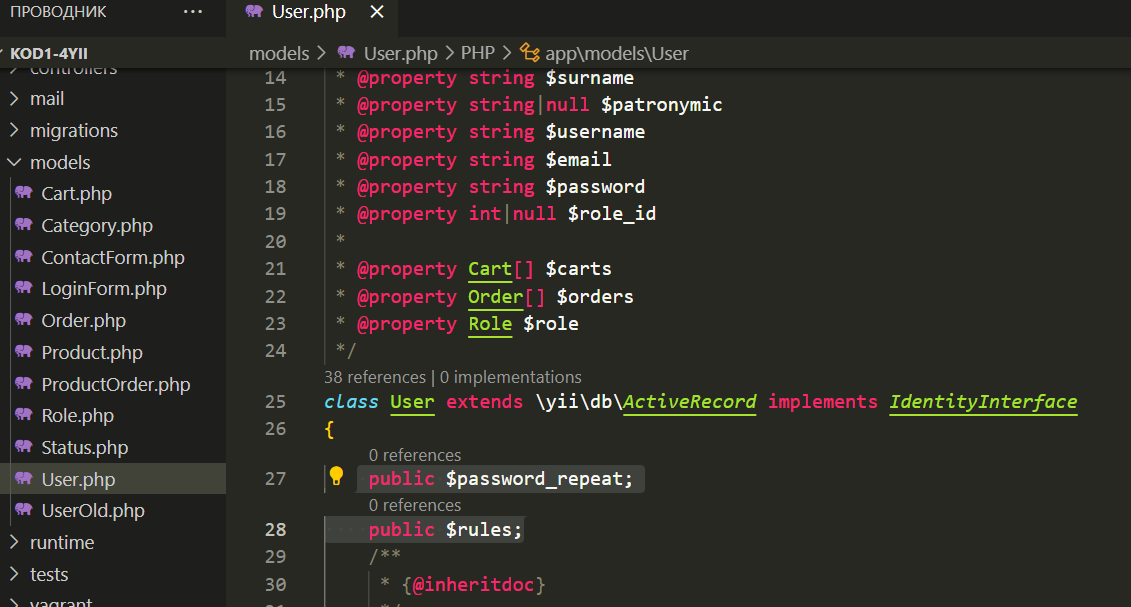
1. Переименовать поля формы регистрации пользователя согласно заданию можно в файле **models\User.php** в методе **attributeLabels()**



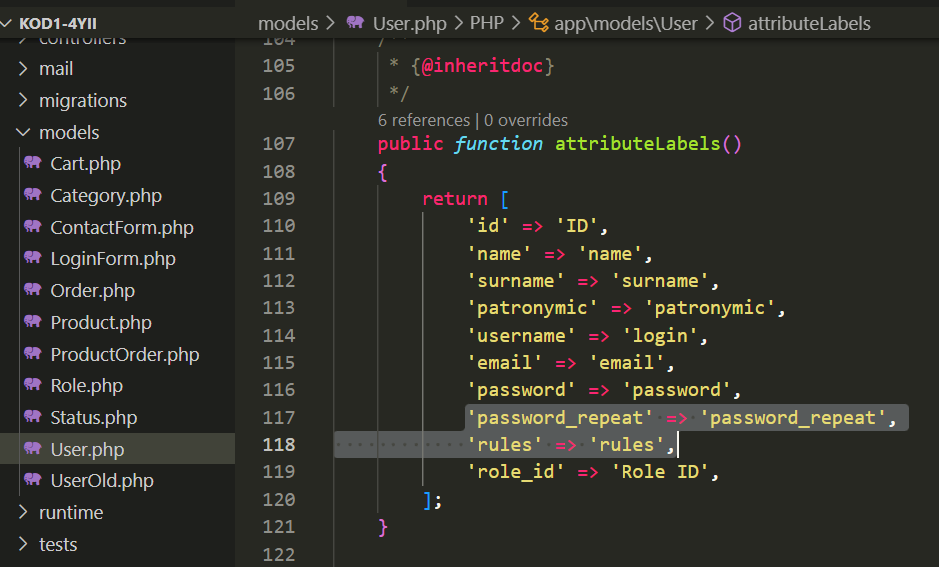
1. Чтобы пользователь не смог добавить себе роль администратора при регистрации в файле **views\user\\_form.php** удалить строку **<?= $form->field($model, 'role\_id')->textInput() ?>**



1. Поля повтор пароля и согласие на обработку персональных данных не должны сохранять данные в базе, поэтому их надо добавить вручную в файл **models\User.php** в класс **User public $password\_repeat;** и **public $rules;** (чтобы не ошибиться названия полей копируем из задания)



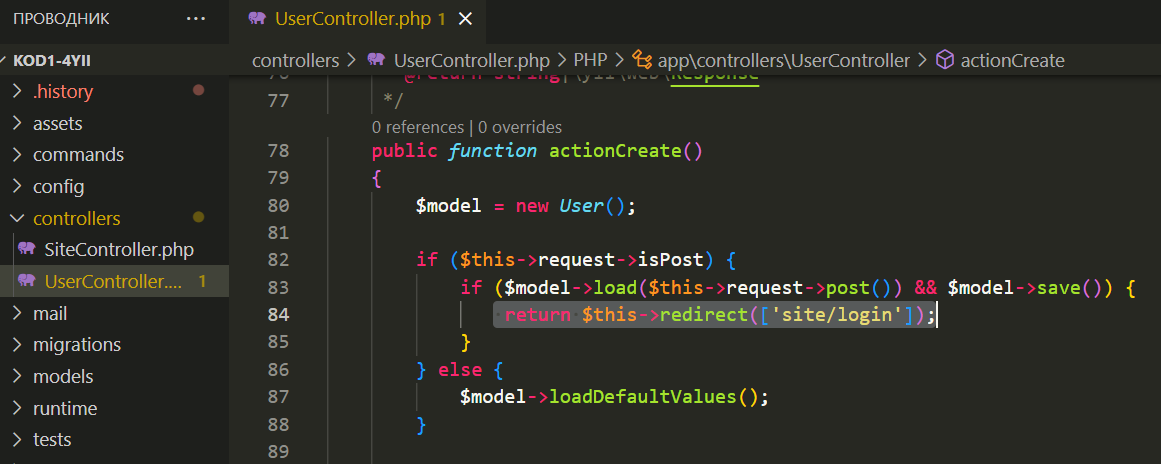
И добавляем их в этот же файл **models\User.php** в метод **attributeLabels()** для отображения в форме



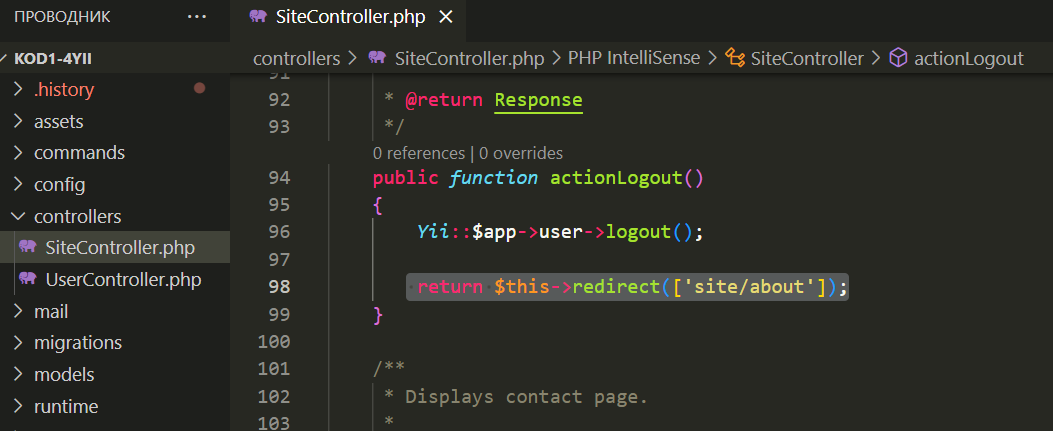
1. Для того, чтобы после успешной регистрации отменить редирект на несуществующую страницу в файле **controllers\UseController.php** в методе **actionCreate()** необходимо исправить одну строку **return $this->redirect(['view', 'id' => $model->id]);** на **return $this->redirect(['site/login']);**



1. Функция выхода из учетной записи должна перенаправлять на страницу **О нас**. Для этого в файле **controllers\UseController.php** в методе **actionCreate()** копируем строчку редиректа **return $this->redirect(['site/login']);**



И вставляем в файл **controllers\SiteController.php** в метод **actionLogout()** и исправляем строку на страницу **site/about**



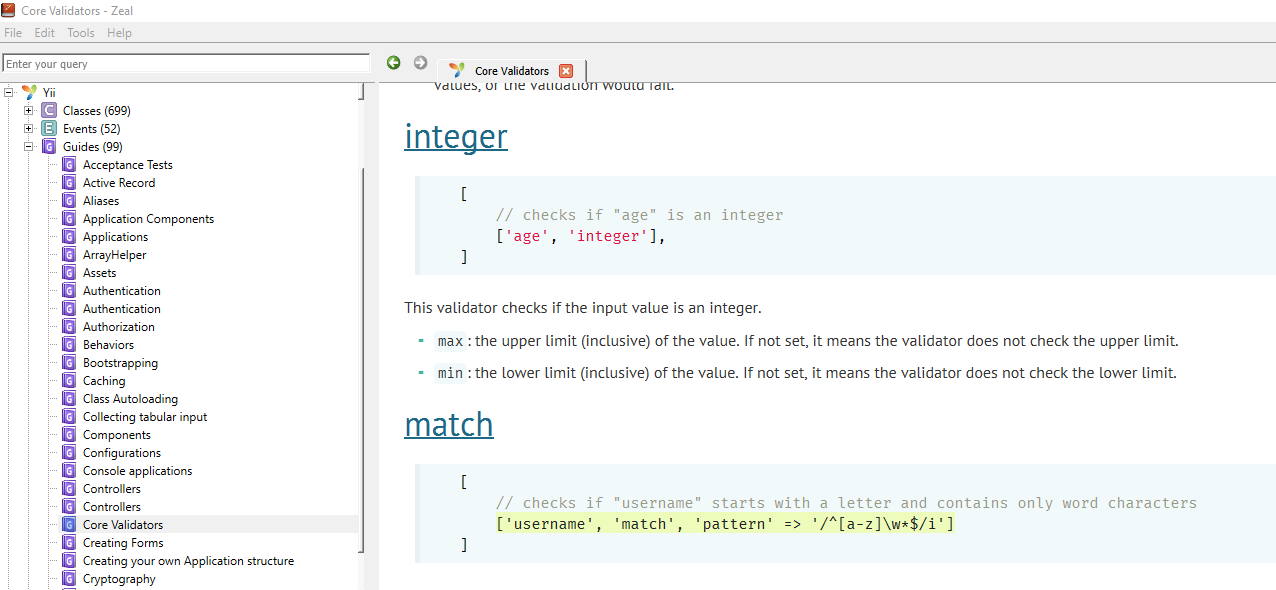
**Валидация**

1. name – обязательное поле, разрешенные символы (кириллица, пробел и тире);

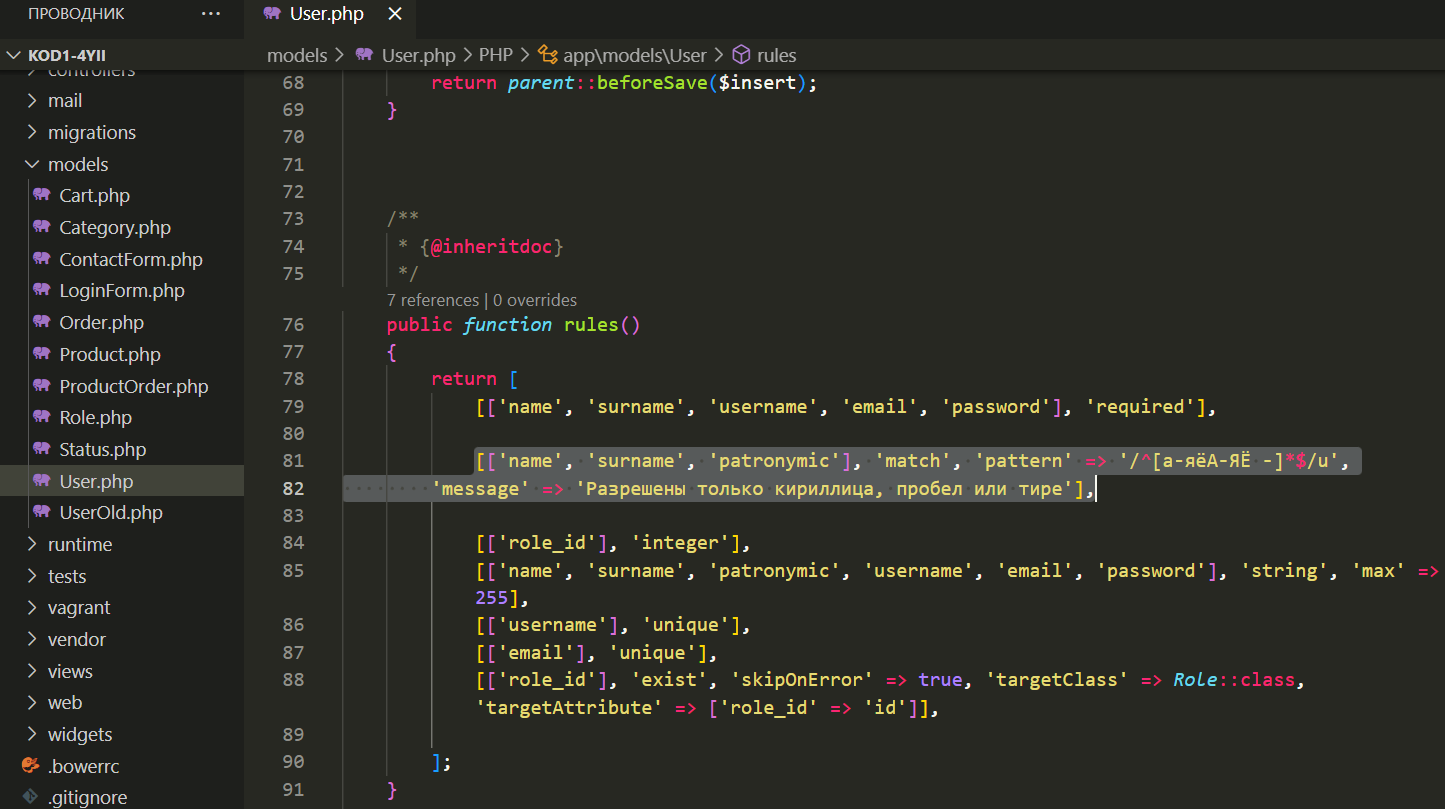
surname – обязательное поле, разрешенные символы (кириллица, пробел и тире);

patronymic – не обязательное поле, разрешенные символы (кириллица, пробел и тире);

Для реализации этих пунктов копируем из **ZEAL Yii -> Guides -> Core Validator** ближе к концу статьи **['username', 'match', 'pattern' => '/^[a-z]\w\*$/i']**

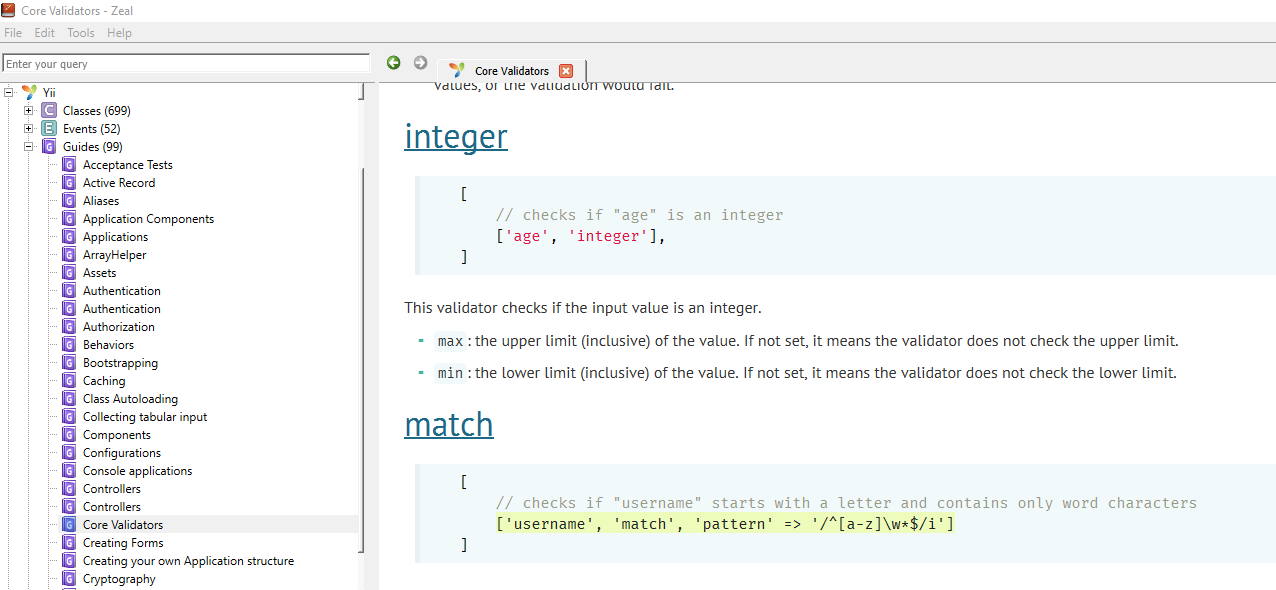


И вставляем в файл **models\User.php** в **метод rules()** под строкой **[['name', 'surname', 'username', 'email', 'password'], 'required'],** и исправим в соответствие с требованием задания **[['name', 'surname', 'patronymic'], 'match', 'pattern' => '/^[а-яёА-ЯЁ -]\*$/u', 'message' => 'Разрешены только кириллица, пробел или тире'],**

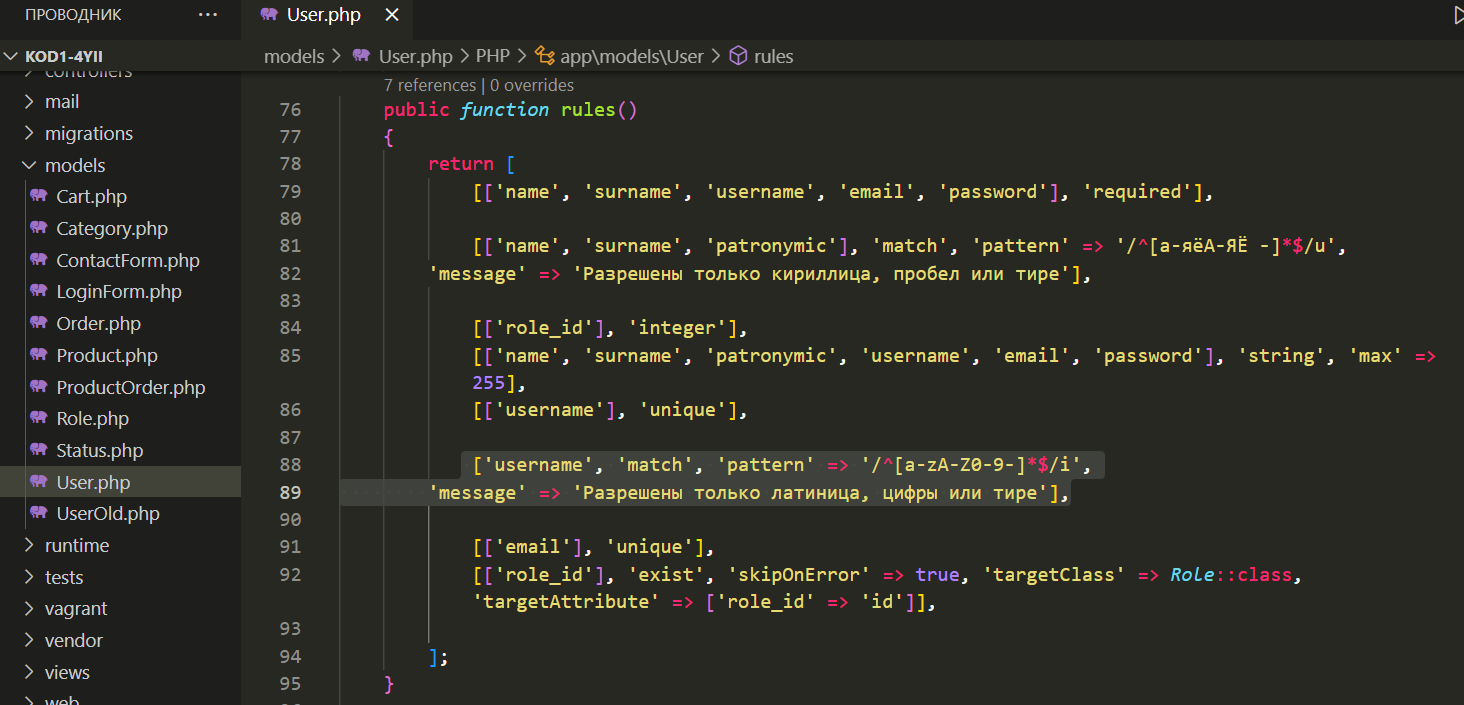


2) login – обязательное и уникальное поле, разрешенные символы (латиница, цифры и тире);

Для реализации этих пунктов копируем из **ZEAL Yii -> Guides -> Core Validator** ближе к концу статьи **['username', 'match', 'pattern' => '/^[a-z]\w\*$/i']**



И вставляем в файл **models\User.php** в **метод rules()** под строкой **[['username'], 'unique'],** и исправим в соответствие с требованием **['username', 'match', 'pattern' => '/^[a-zA-Z0-9-]\*$/i', 'message' => 'Разрешены только латиница, цифры или тире'],**

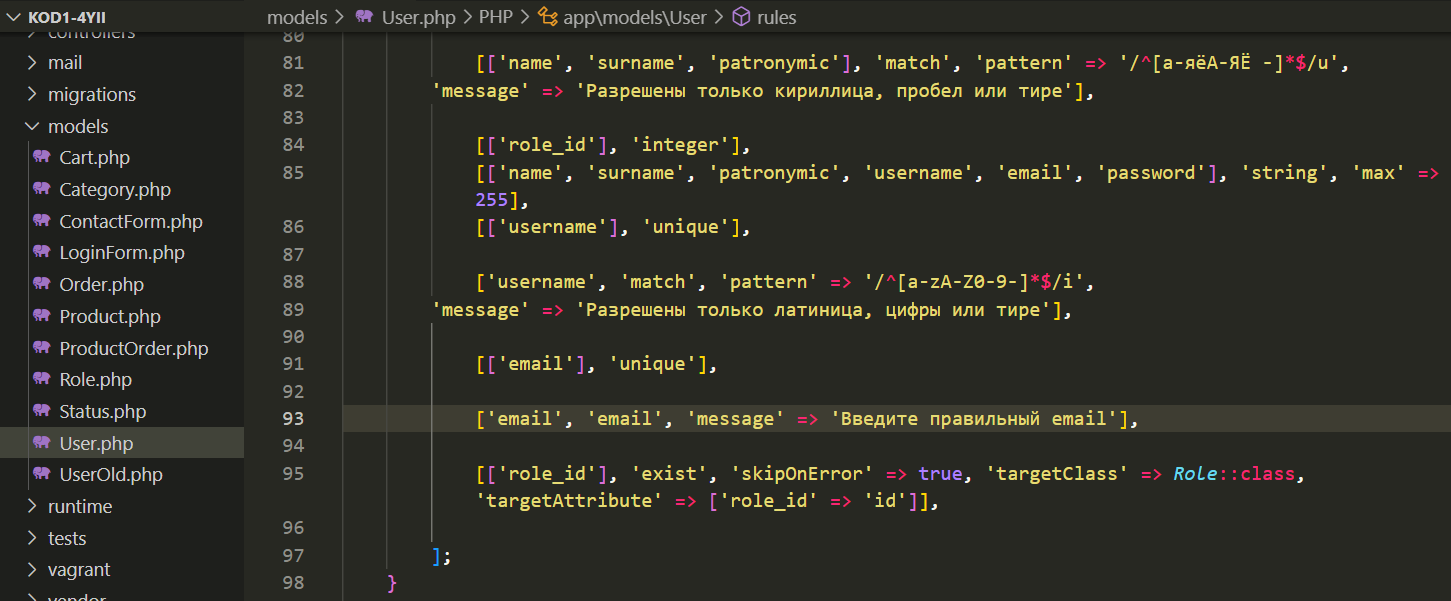


3) email – обязательное и уникальное поле, проверка на email;

Для реализации этого пункта копируем из **ZEAL Yii -> Guides -> Core Validator** ближе к началу статьи **['email', 'email'],**

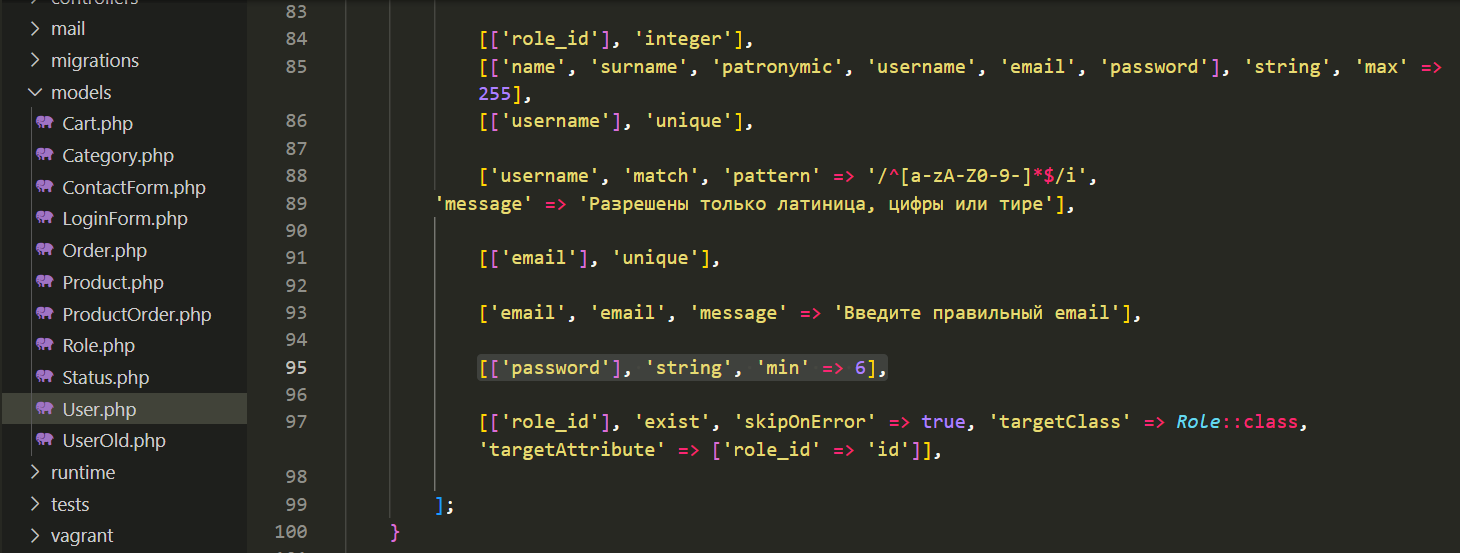


И вставляем в файл **models\User.php в метод rules()** под строкой **[['email'], 'unique'],** и исправим в соответствие с требованием **['email', 'email', 'message' => 'Введите правильный email'],**

****

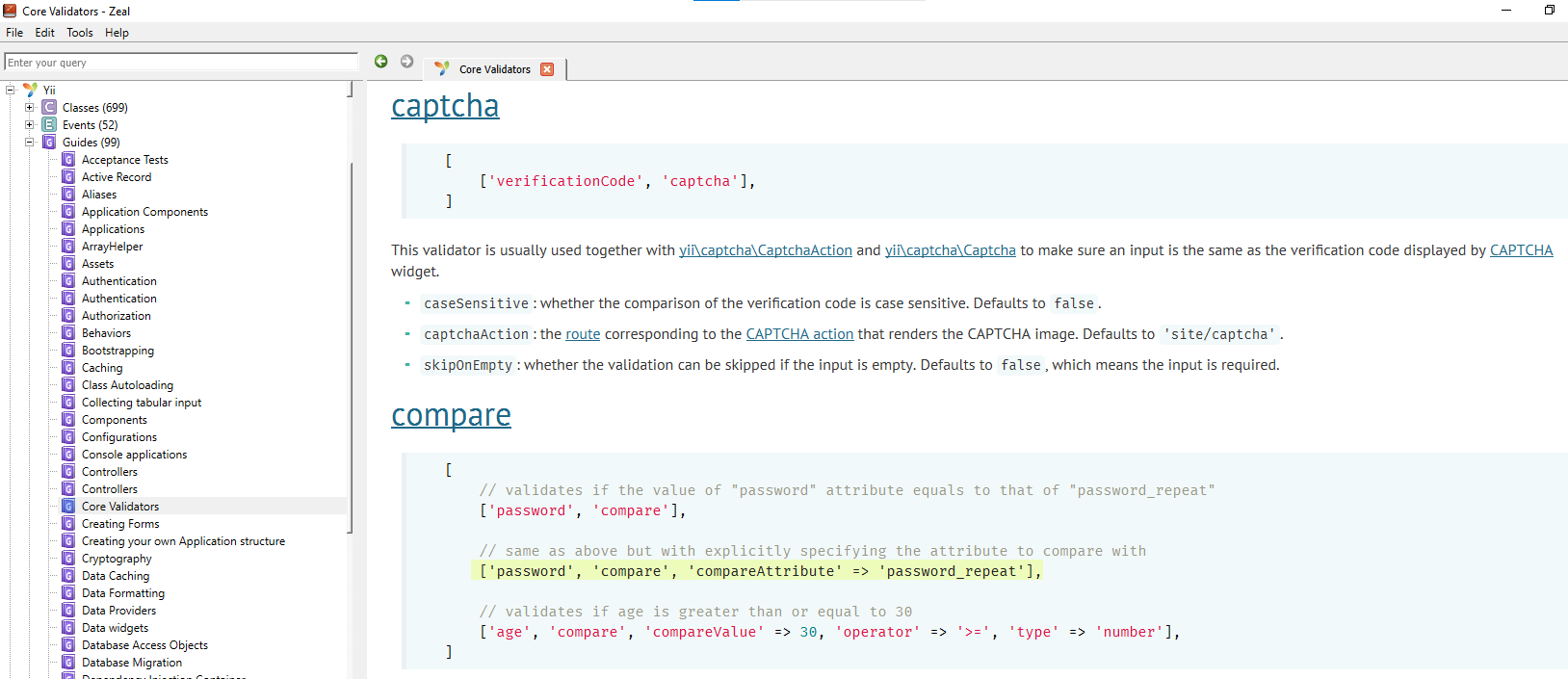
4) password – обязательное поле, не менее 6-ти символов;

Для этого исправим строку [['password'], 'string'], в том же файле **models\User.php в метод rules()** на **[['password'], 'string', 'min' => 6],**



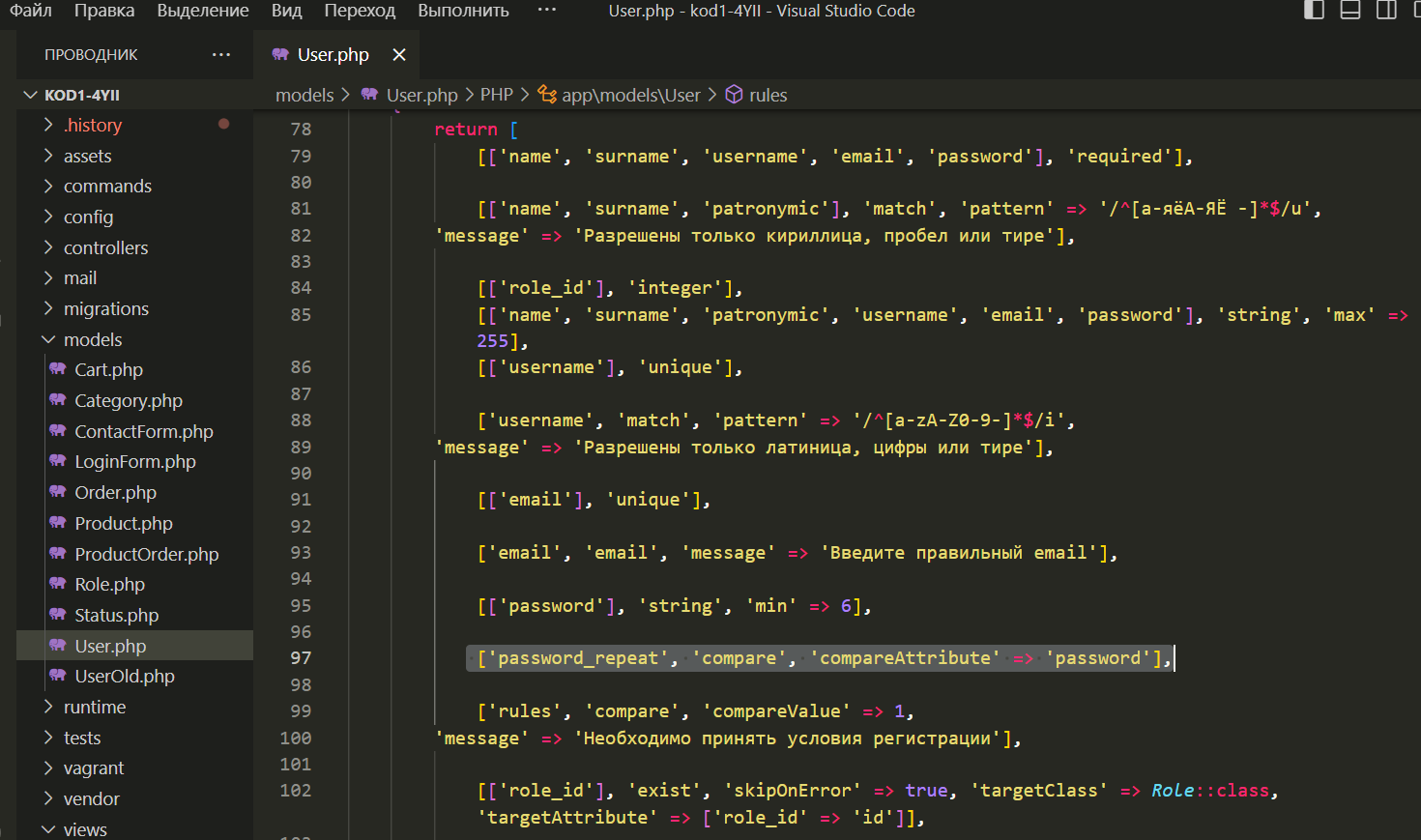
5) password\_repeat – обязательное поле, должно совпадать с полем password;

Для реализации этого пункта копируем из **ZEAL Yii -> Guides -> Core Validator** ближе к началу статьи **['password', 'compare', 'compareAttribute' => 'password\_repeat'],**

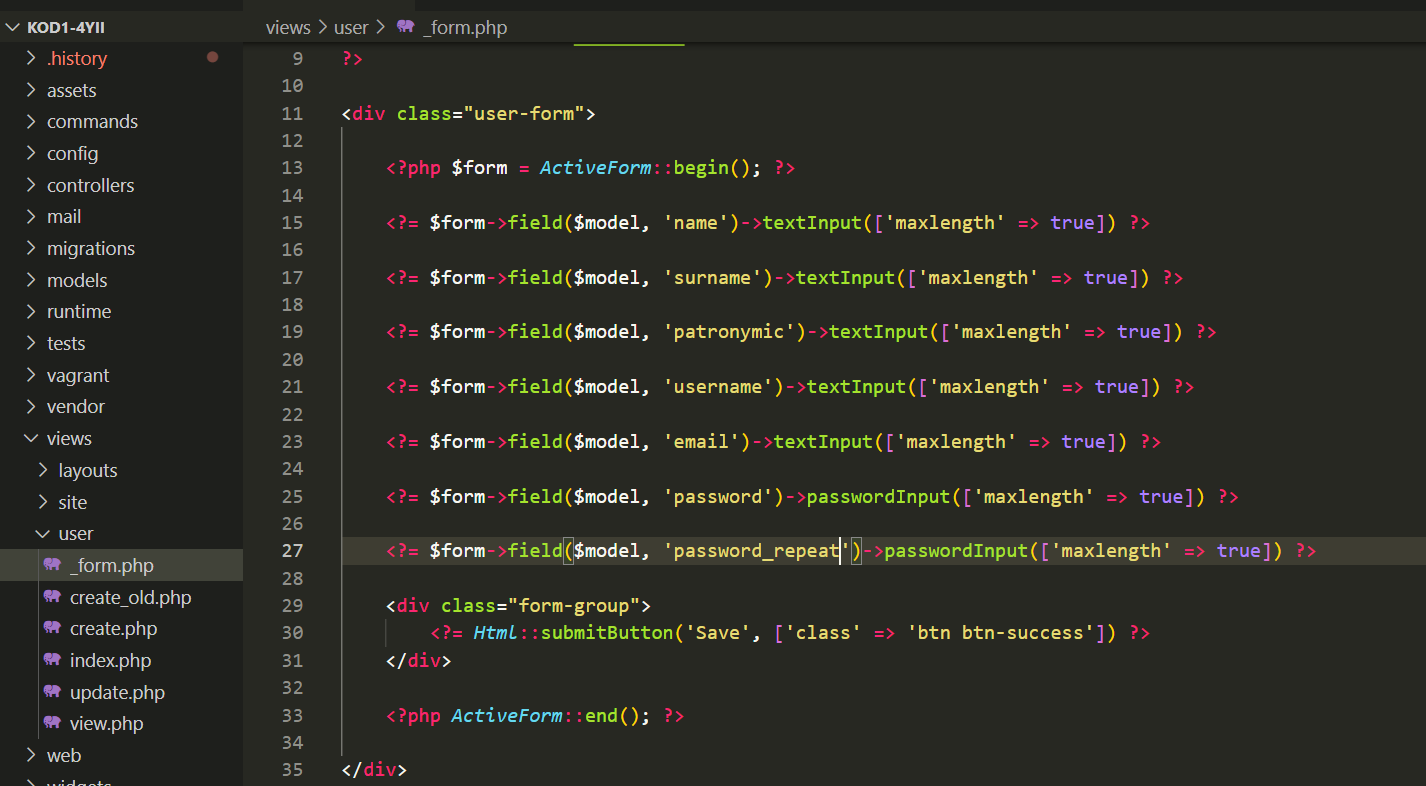


И вставляем в файл **models\User.php в метод rules()** под строкой **[['password'], 'string', 'min' => 6],**

**['password', 'compare', 'compareAttribute' => 'password\_repeat'],** и исправляем на **['password\_repeat', 'compare', 'compareAttribute' => 'password'],**



6) Для добавления поля повтора пароля в форму регистрации, надо в файле **views\user\\_form.php** в **<div class="user-form">** скопировать строку **<?= $form->field($model, 'password')->passwordInput(['maxlength' => true]) ?>** и вставить её ниже скопированной, а затем исправить **password** на **'password\_repeat'**

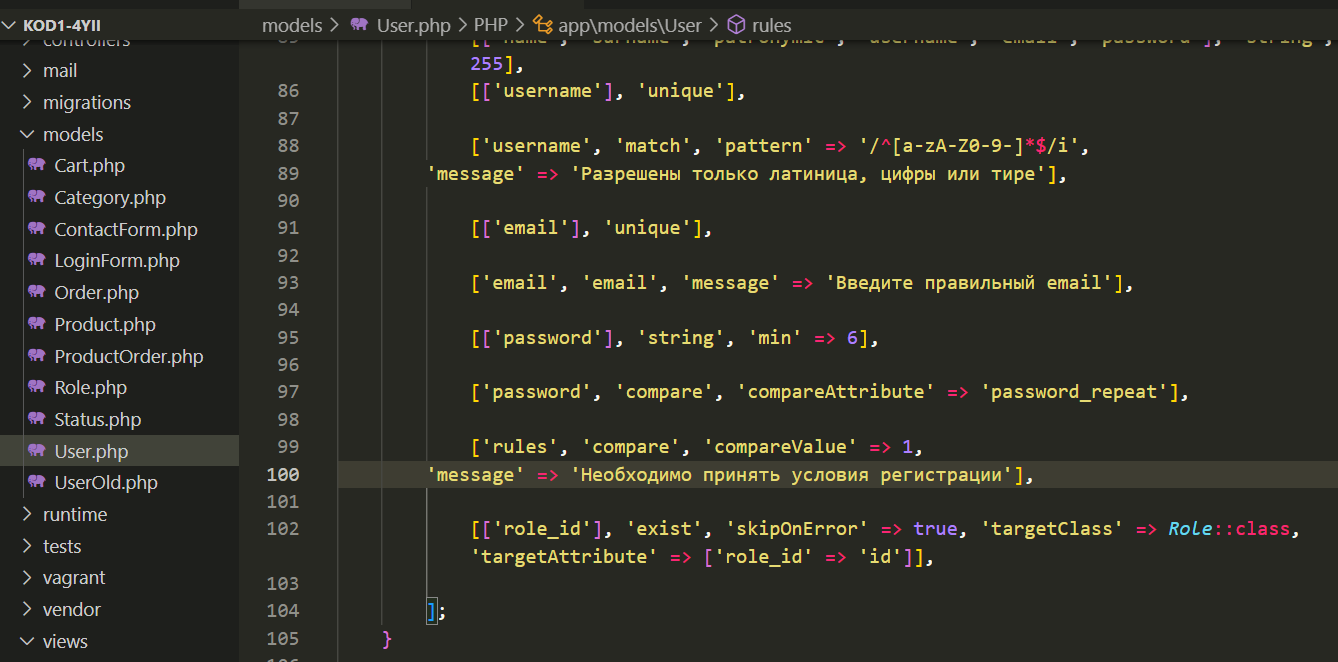


7) Для добавления поля rules - согласие с правилами регистрации в форму регистрации надо в файле **views\user\\_form.php в <div class="user-form">** скопировать строку **<?= $form->field($model, 'password\_repeat')->passwordInput(['maxlength' => true]) ?>** и вставить её ниже скопированной, а затем исправить **'password\_repeat’** на **'rules'** и **passwordInput(['maxlength' => true])** на **checkbox()**



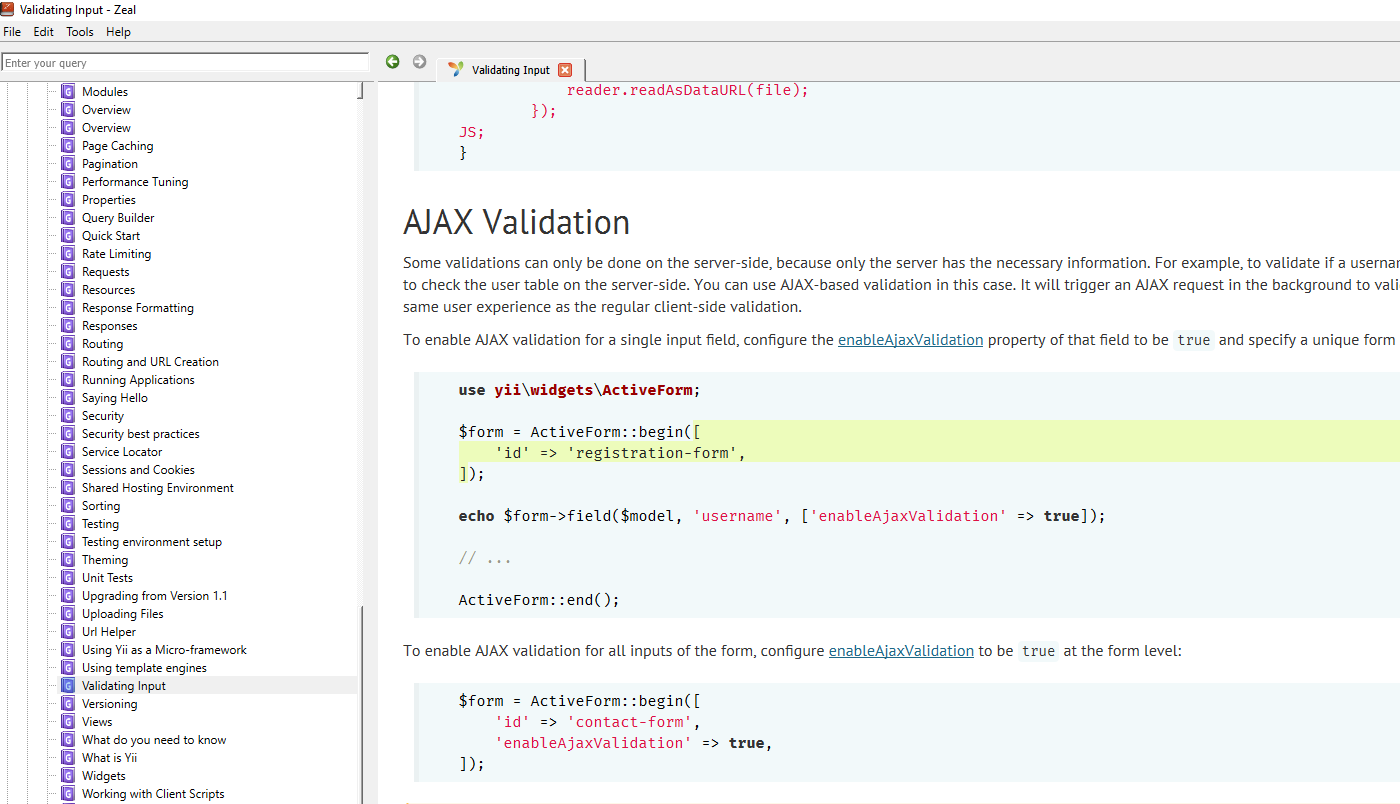
8) rules - согласие с правилами регистрации.

Для реализации этого пункта копируем из **models\User.php** строку **['password', 'compare', 'compareAttribute' => 'password\_repeat'],** и вставляем её ниже скопированной строки. Затем исправляем **['rules', 'compare', 'compareValue' => 1, 'message' => 'Необходимо принять условия регистрации'],**

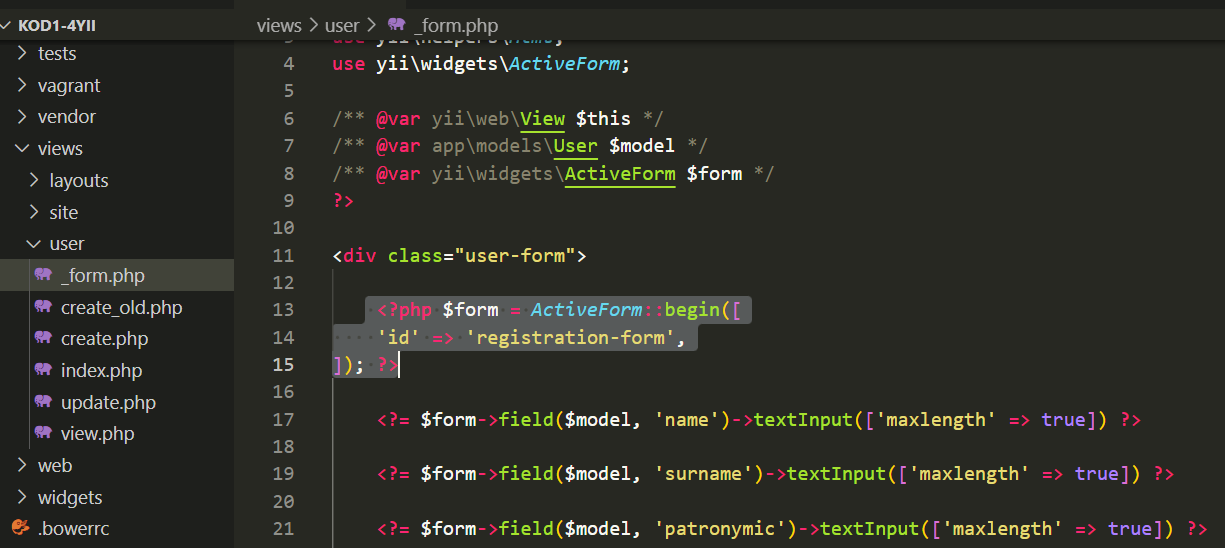


Проверяем в браузере, что валидация работает

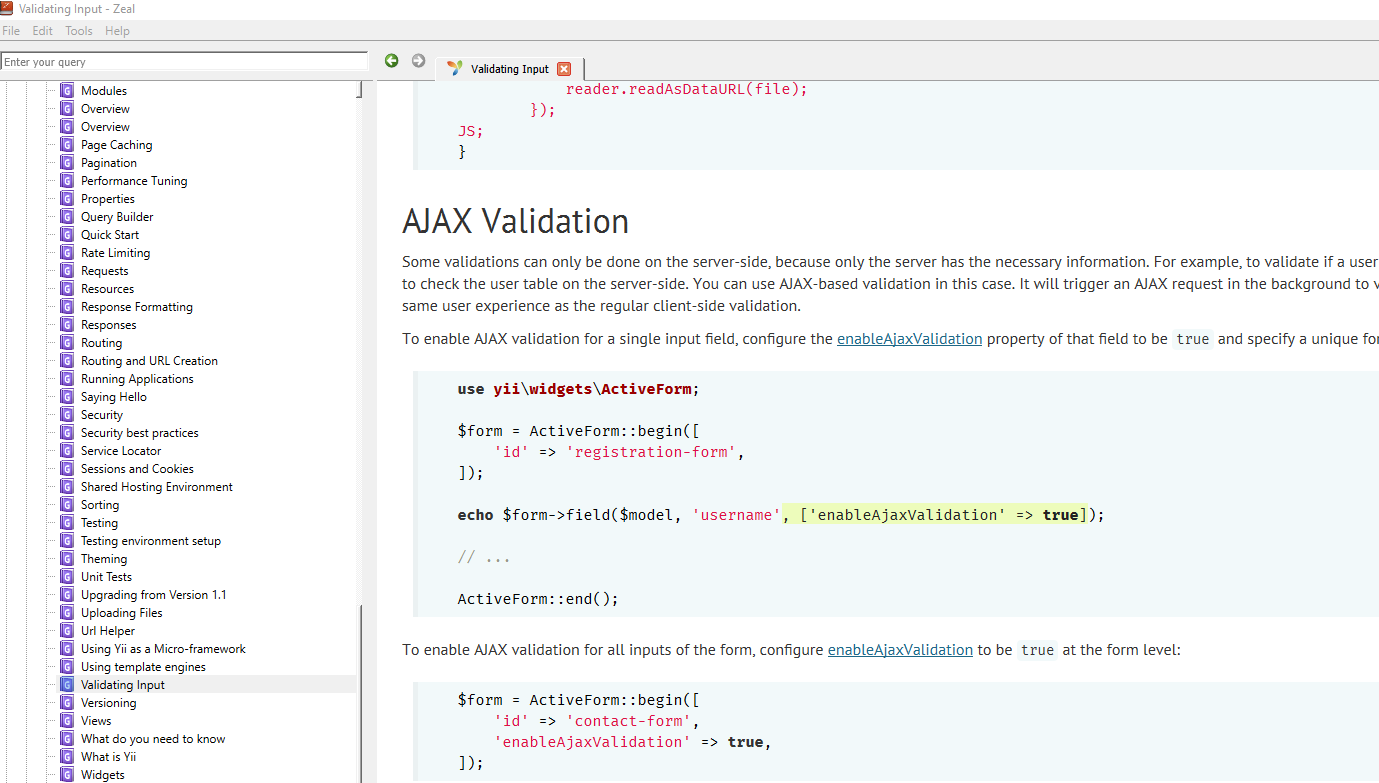
9) login – обязательное и уникальное поле, email – обязательное и уникальное поле. Чтобы сделать проверку на уникальность без перезагрузки страницы необходимо использовать технологию Ajax. Для этого в **ZEAL Yii -> Validating Input** ближе к концу в статье **AJAX Validation** копируем строку **['id' => 'registration-form',**]



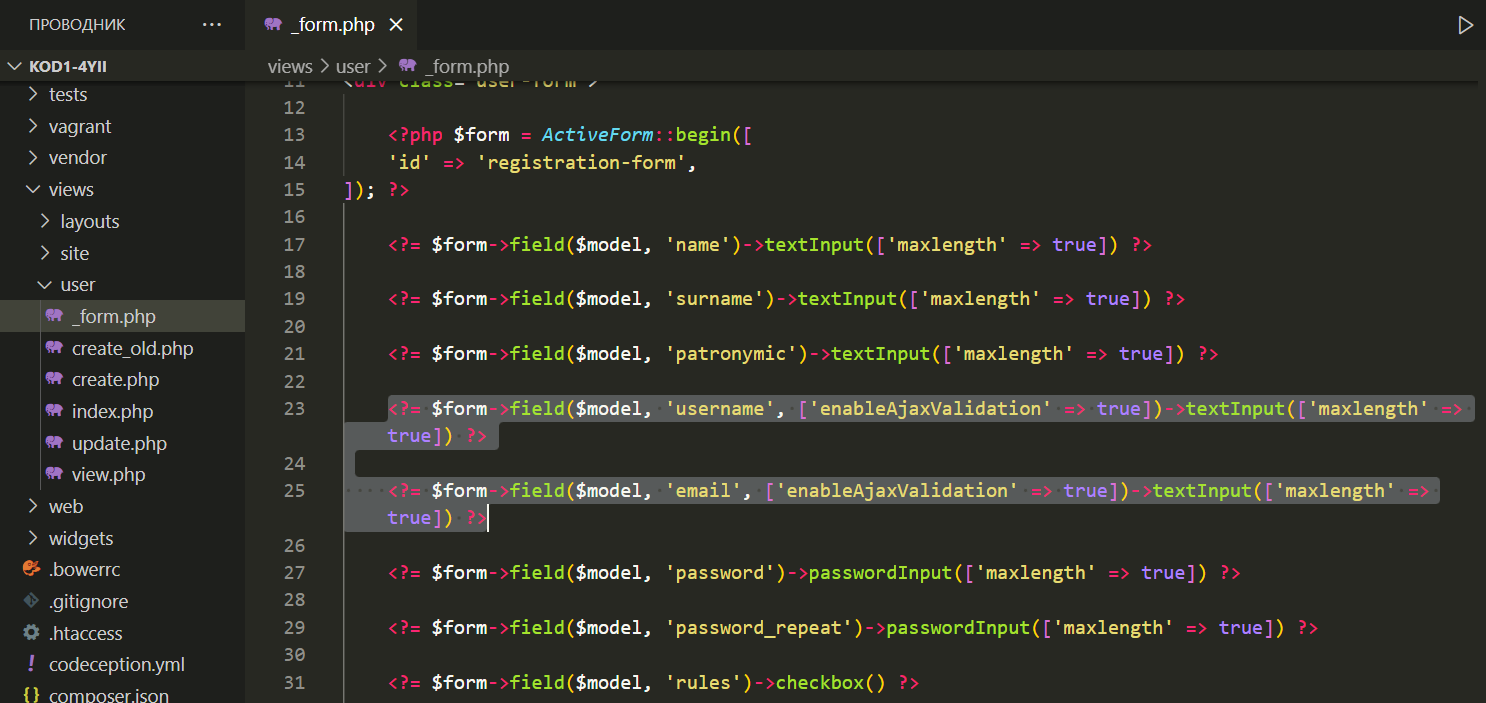
Вставить в файл **views\user\\_form.php** в метод **ActiveForm::begin()**



10) В **ZEAL Yii -> Validating Input** ближе к концу в статье **AJAX Validation** копируем строку, **['enableAjaxValidation' => true]**



Вставить в файл **views\user\\_form.php** в метод **ActiveForm::begin(['id' => 'registration-form',])** в строки, **username** и **email**

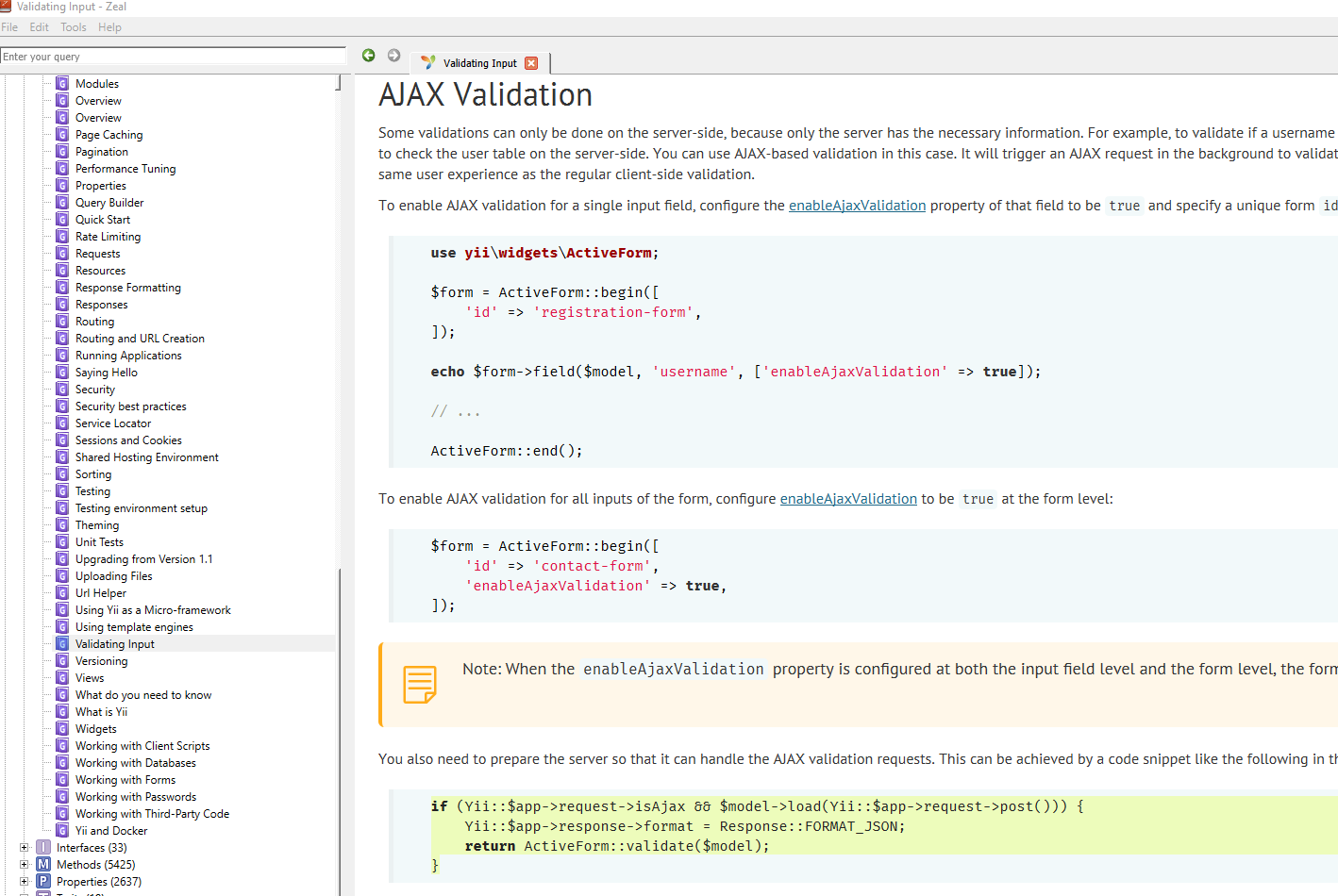


11) В **ZEAL Yii -> Validating Input** ближе к концу в статье **AJAX Validation** копируем строки,

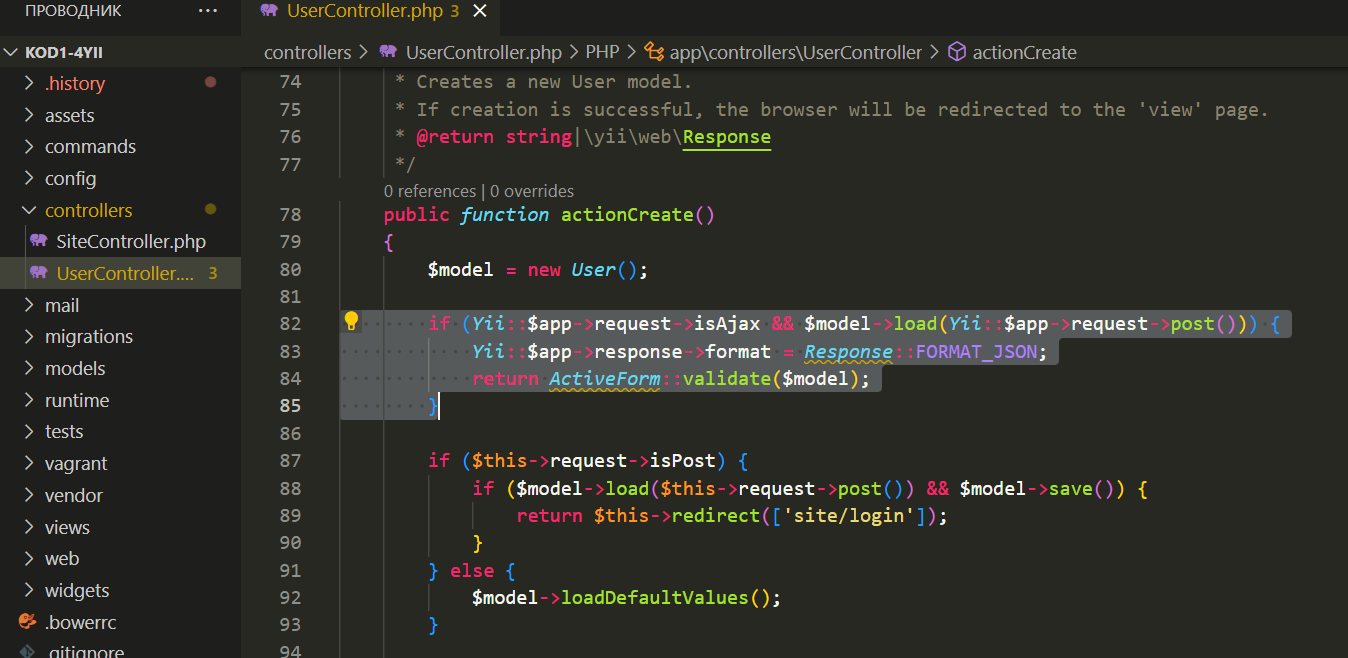
if (Yii::$app->request->isAjax && $model->load(Yii::$app->request->post())) {

Yii::$app->response->format = Response::FORMAT\_JSON;

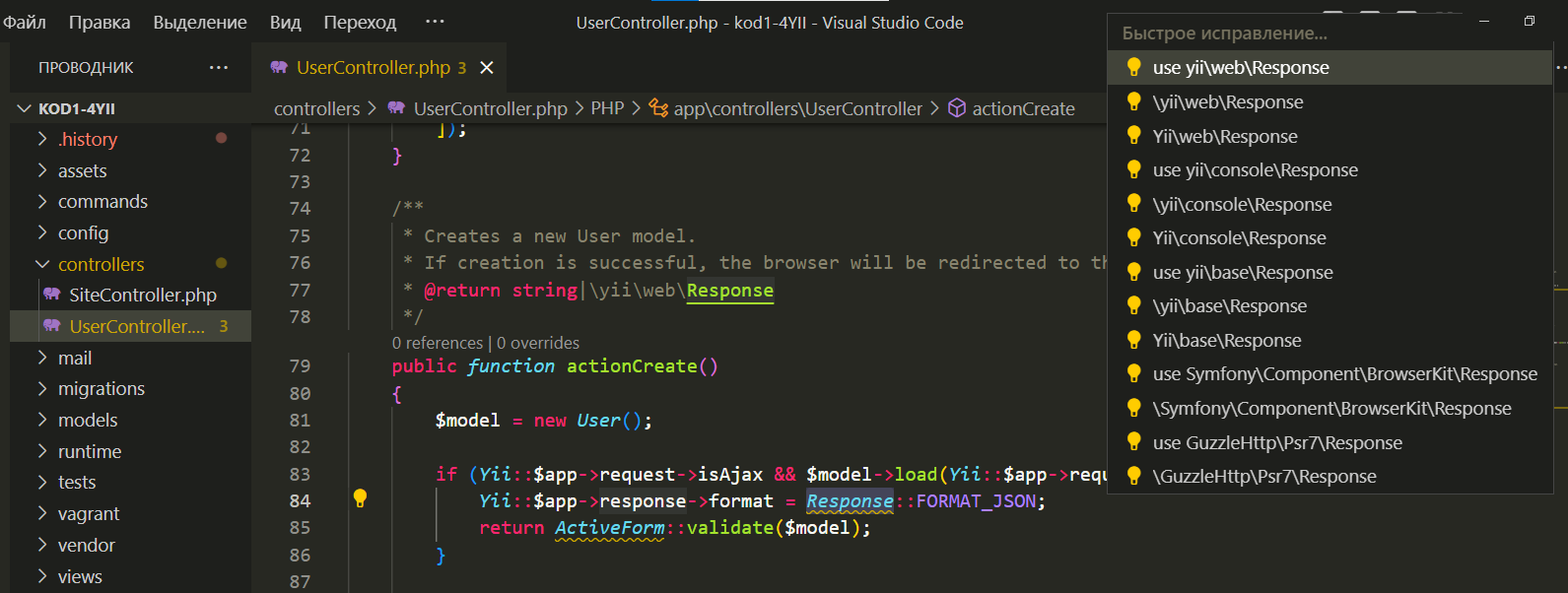
return ActiveForm::validate($model);}



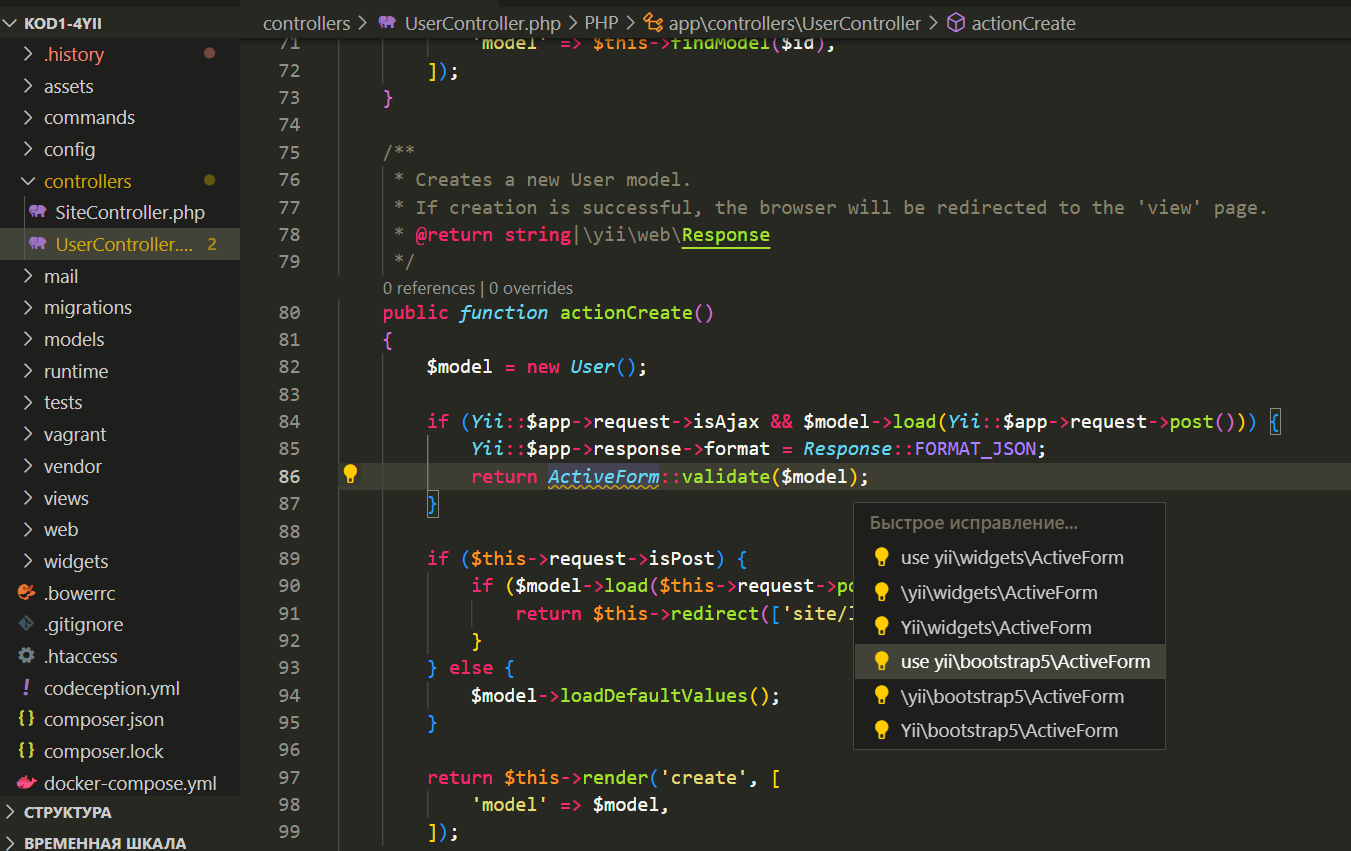
Вставить в файл **controllers\UserController.php** в метод **actionCreate()** под строку **$model = new User();**



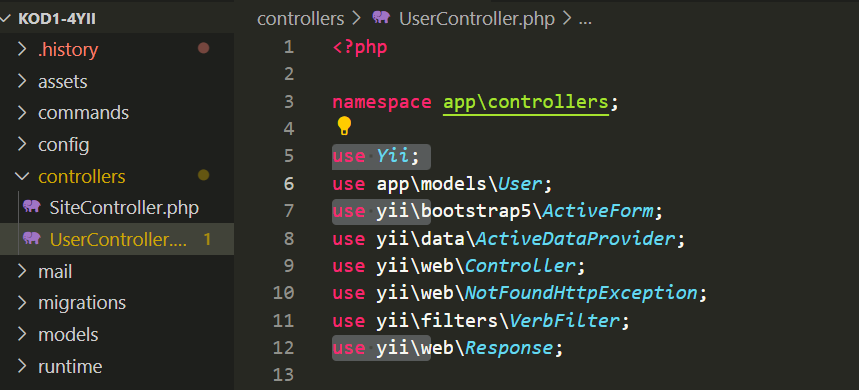
12) Во вставленном фрагменте наведите по очереди мышкой на подчеркнутые слова **Response** и **AtiveForm** и выберите в модальном окне Быстрое исправление. В следующем модальном окне выберите **use yii\web\Response** для слова **Response**



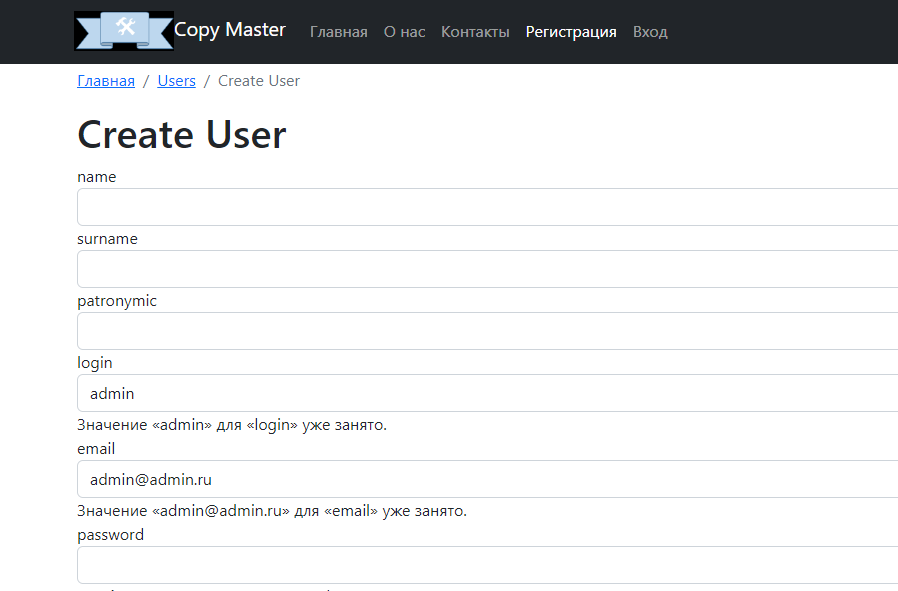
И **use yii\bootstrap5\ActiveForm;** для слова **AtiveForm**



13) Добавить класс **Yii** в **controllers\UserController.php** и проверьте, что в объявление классов добавлены классы из предыдущих пунктов



14) Проверим как работает валидация через AJAX. Для этого в браузере на странице Регистрация в поле **login** и **email** вводим существующие в базе данных значения **login – admin email – admin@admin.ru**. Если всё сделано правильно, то без перезагрузки страницы мы увидим сообщение **Значение «admin» для «login» уже занято** и **Значение «admin@admin.ru» для «email» уже занято**



По критериям оценки выполненное задание составляет:

Часть 1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| O | Сайт доступен по заданному адресу |  |  | 1 | 0,25 |
| O | Администратор добавлен в систему (можно авторизоваться) |  | admin - admin00 | 1 | 0,25 |
| O | На сайте представлен логотип |  |  | 3 | 0,25 |
| O | Сверстана кнопка или ссылка Выход |  |  | 4 | 0,20 |
| O | Сверстана страница Авторизации |  |  | 4 | 0,50 |
| O | Реализована авторизация пользователей |  |  | 6 | 0,50 |
| O | При не правильной паре логин/пароль выводится сообщение |  |  | 6 | 0,25 |
| O | Реализован выход |  |  | 6 | 0,25 |
| O | Отсутсвуют ошибки, связанные с безопасностью |  | mysql инъекции, XSS-атаки | 6 | 1,00 |

Часть 2

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| O | Осмысленные имена функций и переменных |  |  | 2 | 0,50 |
| J | Качество PHP кода | 3 | Используется PHP framework, код организован и легко читаем | 2 | 0,75 |
| J | Использование комментариев в коде | 3 | Используются осмысленные вводные и пояснительные комментарии | 2 | 0,25 |
| J | Структура веб проекта разработана качественно | 3 | Структура файлов позволяет без проблем найти необходимые данные | 2 | 0,50 |
| J | Качество реализации логотипа | 3 | Профессиональная реализация работы | 3 | 1,50 |
| J | Удобство интерфейса | 3 | Крайне хорошо проработанный интерфейс, использовать удобно | 3 | 1,00 |
| J | Присутствует адаптивная верстка | 3 | Полностью учитываются особенности мобильных платформ | 3 | 1,00 |
| O | Валидность HTML кода |  | Штраф - 0,25 за каждый тип ошибки | 4 | 0,75 |
| O | Валидность CSS кода |  | Штраф - 0,5 за каждый тип ошибки | 4 | 1,50 |
| O | Сверстана страница Регистрации |  | штраф за каждый отсутсвующий элемент 0,2 | 4 | 0,80 |
| O | В форме регистрации поля ввода e-mail имеет корректный тип |  | или проводится валидация адреса email средствами js | 4 | 0,20 |
| O | Проверка валидации полей ФИО выполняется |  |  | 5 | 0,75 |
| O | Форма регистрации не отправляется, если данные не заполнены |  |  | 5 | 0,50 |
| O | Проверка на уникальность логина проходит без перезагрузки страницы |  |  | 5 | 0,75 |
| O | Проверка повтора пароля |  |  | 5 | 0,25 |
| O | Проверка согласия с правилами регистрации |  |  | 5 | 0,25 |
| O | Валидации формы регистрации проходят без перезагрузки страницы |  |  | 5 | 0,50 |
| O | Данные сохраняются в БД |  |  | 6 | 0,50 |
| O | Реализована регистрация |  |  | 6 | 0,75 |
| O | Проверка валидности логина |  |  | 6 | 0,25 |
| O | Проверка уникальности логина |  |  | 6 | 0,50 |

Итого 17,2 балла или 39,1% задания.

Схема перевода результатов демонстрационного экзамена из сто балльной шкалы в пятибалльную

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Оценка** | **«2»** | **«3»** | **«4»** | **«5»** |
| Отношение полученного количества баллов к максимально возможному (в процентах) | **0,00% - 19,99%** | **20,00% - 39,99%** | **40,00% - 69,99%** | **70,00% - 100,00%** |