**Оформление**

1. **Логотип**

Чтобы изменить логотип на свой, в config/web добавим строчку:

'name'=>'MyForum',

Получится:

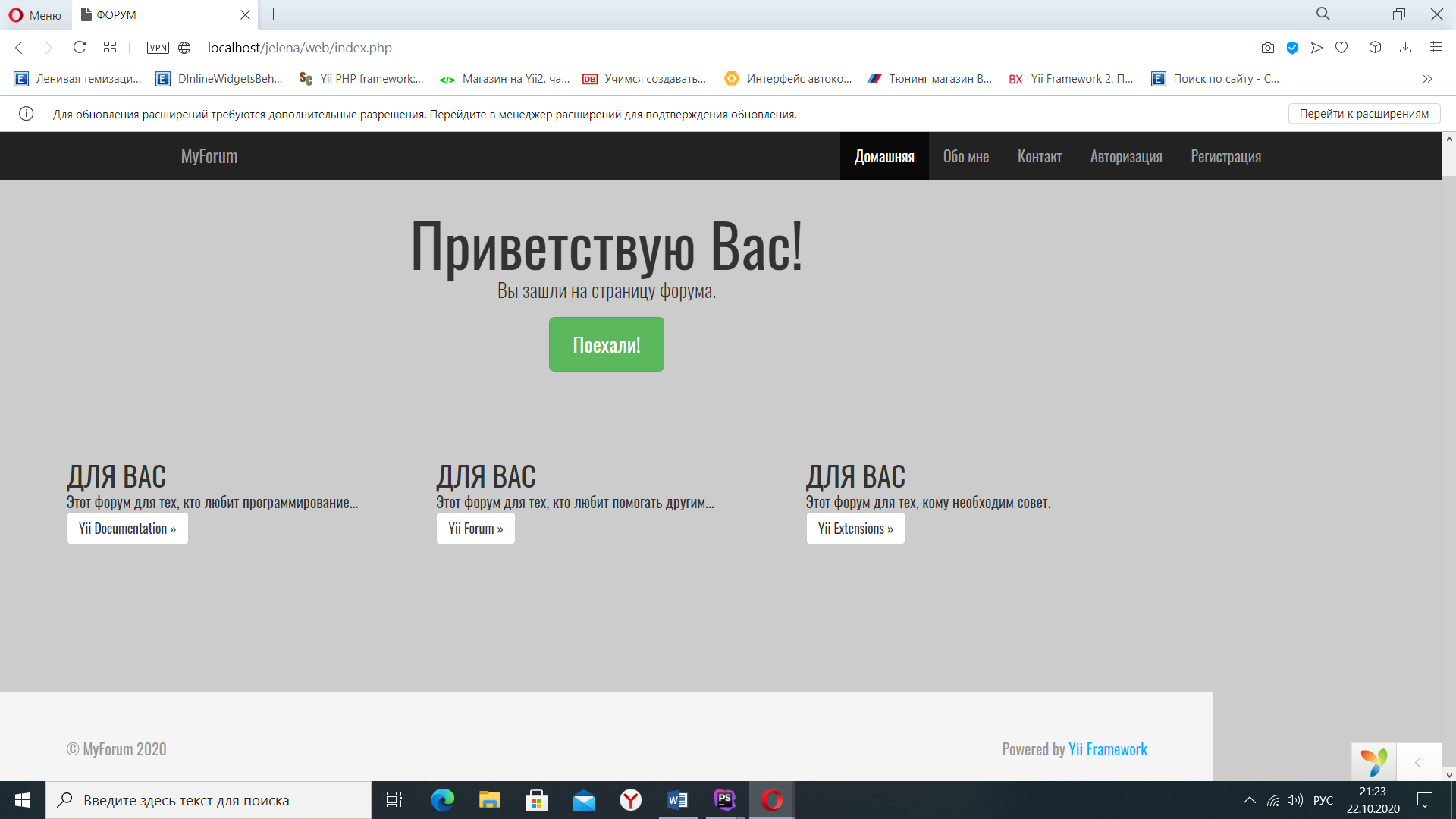
$config = [  
 'id' => 'basic',  
 'name'=>'MyForum',  
 'basePath' => *dirname*(*\_\_DIR\_\_*),  
 'bootstrap' => ['log'],

Логотип поменялся.

В футере шаблона (у меня шаблон main)тоже можно поменять MyCompany на свое:

<footer class="footer">  
 <div class="container">  
 <p class="pull-left">&copy; MyForum <?= *date*('Y') ?></p>

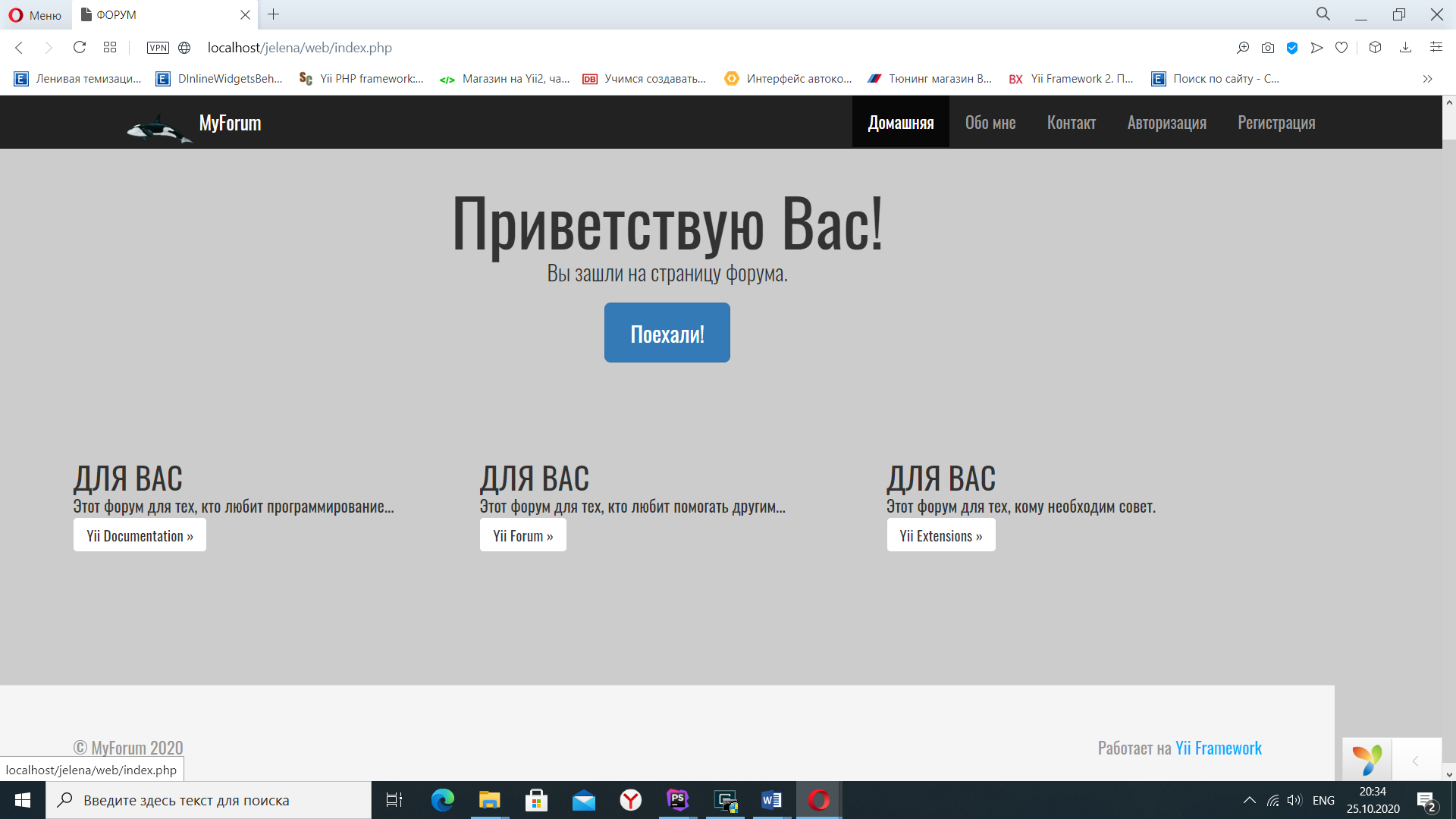
….



Чтобы добавить логотип в виде изображение в поле заголовка , надо положить картинку с логотипом в папку web/images и вписать в шаблон после NavBar::begin([ следующий код:

NavBar::*begin*([  
 'brandLabel' => '<img src="images/whale.png" style="display: inline; vertical-align: top; height:32px;"> MyForum',

На скриншоте ниже Вы видите картинку с изображением касатки, добавленную перед текстом «MyForum»



1. **Язык интерфейса**

Теперь изменим язык интерфейса: сообщения об ошибках, хлебные крошки и остальное.

Для исправления ситуации можно сразу указать язык приложения language в config/web.php`:

$config = [

...,

'language'=>'ru-RU',

...

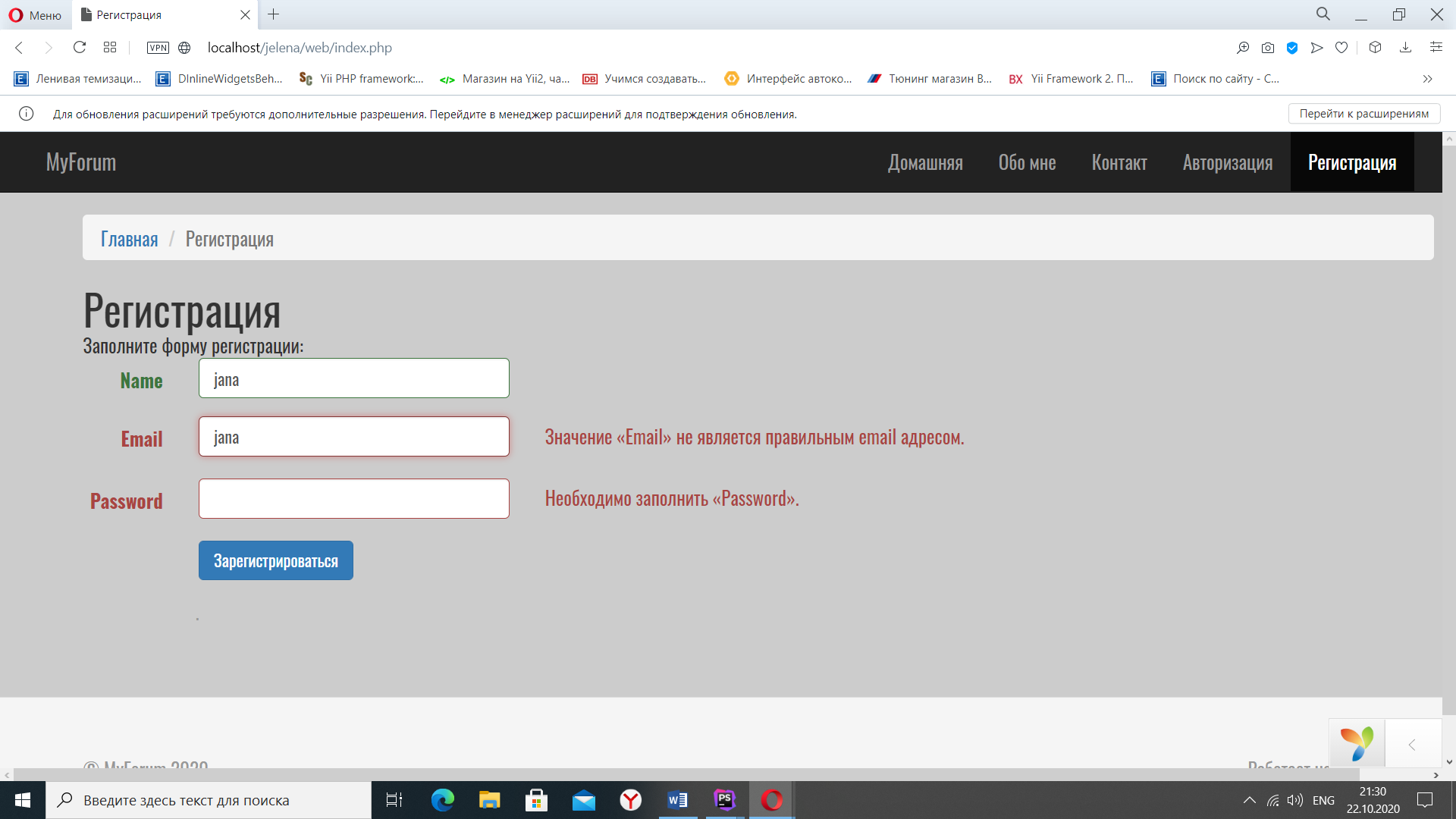
];

Получится:

$config = [  
 'id' => 'basic',  
 'name'=>'MyForum',  
 'language'=>'ru-RU',  
 'basePath' => *dirname*(*\_\_DIR\_\_*),  
 'bootstrap' => ['log'],…

Теперь все системные сообщения фреймворка и ссылка на главную страницу в хлебных крошках будут выводиться на русском:

Попробуем заполнить форму регистрации с ошибками. Все ошибки стали выводиться на русском:



Остались непереведёнными пункты меню, имена полей форм и фрагменты хлебных крошек. Это всё вписано в представления(виды) вручную, поэтому мы можем заменить их сами, как мы это делали в предыдущей работе.

1. **Шаблоны**

Шаблон используемый по умолчанию находится в

/views/layouts/main.php

Сменить шаблон можно разными способами.

1. **Глобальная смена файла общего шаблона** (замена для main.php на новый шаблон basic.php).

Для этого идем в конфигурационный файл

/config/web.php

добавляем в него следующий параметр (шаблон находится в файле /views/layouts/basic.php)

Код:

'layout' => 'basic',

Получится:

$config = [

'id' => 'basic',

'basePath' => dirname(\_\_DIR\_\_),

'bootstrap' => ['log'],

'layout' => 'basic',

'aliases' => [

Теперь вместо шаблона main для всего сайта будет использоваться шаблон basic.

Чаще нам нужно сделать разные шаблоны для разных страниц сайта.

Для этого мы можем прописать шаблон как отдельному контроллеру, так и отдельно любому методу в контроллере.

**2. Смена шаблона для контроллера.**

Открываем необходимый контроллер и прописываем ему:

public $layout = 'basic';

— теперь данный контроллер будет выводиться через шаблон

/views/layouts/basic.php

**3. Смена шаблона в методе (action).**

Просто в метод добавляем параметр

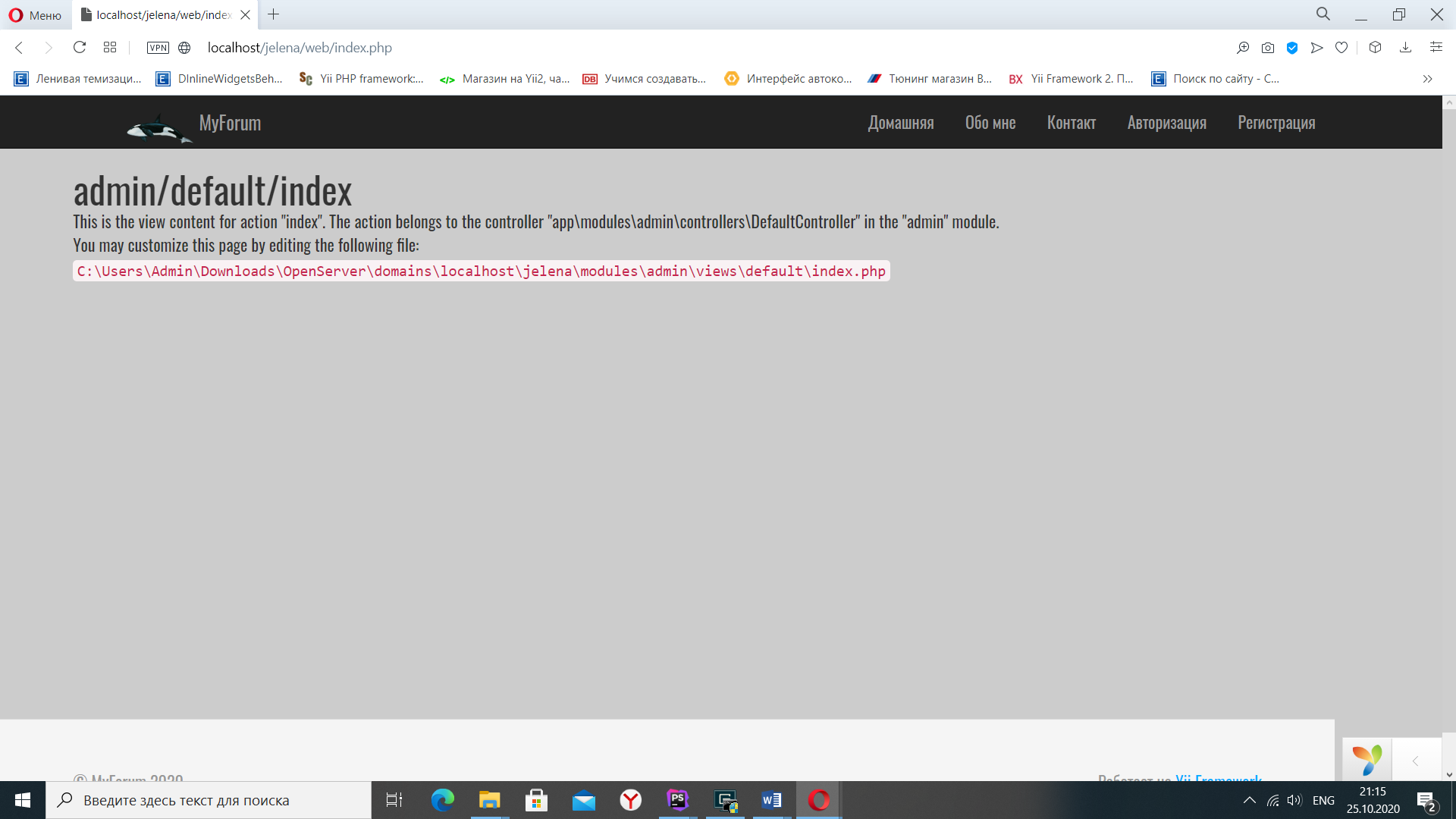
Код:

$this->layout = 'basic';

— теперь данный шаблон будет только для данной страницы

Рассмотрите все эти способы подключения шаблонов и установите три разных шаблона: для админки, для остальной части сайта, для нашей страницы country. Смена шаблона может пригодиться не только, чтобы по-разному оформить страницы, но и чтобы в разных разделах сайта вывести разное меню.

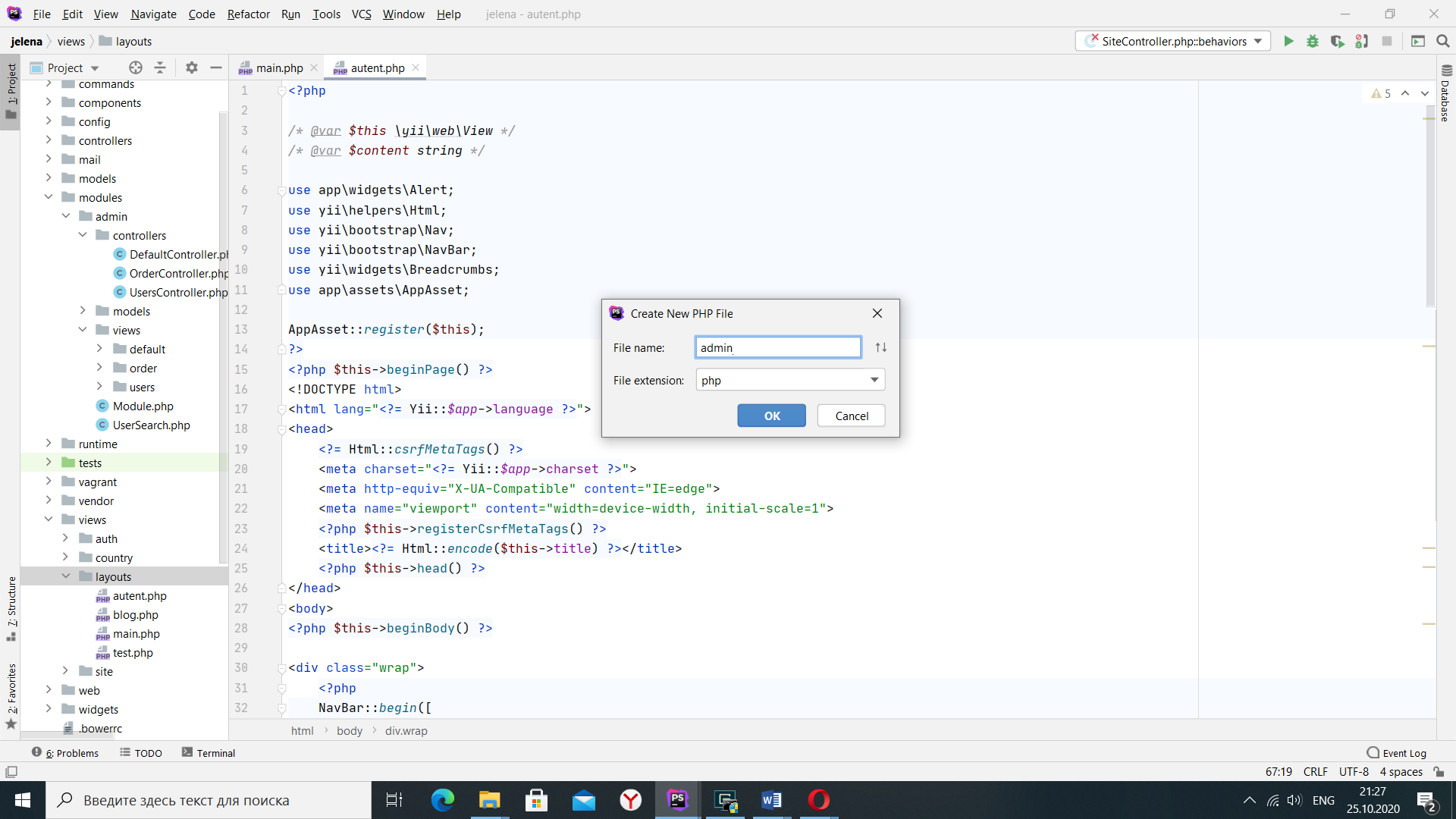
Например, у меня для всего сайта установлен шаблон с меню, в котором 5 закладок (пунктов). Зайду в админку, там тоже 5 закладок:



а я хочу, чтобы пунктов обо мне, контакт, регистрация и авторизация в админке не было.

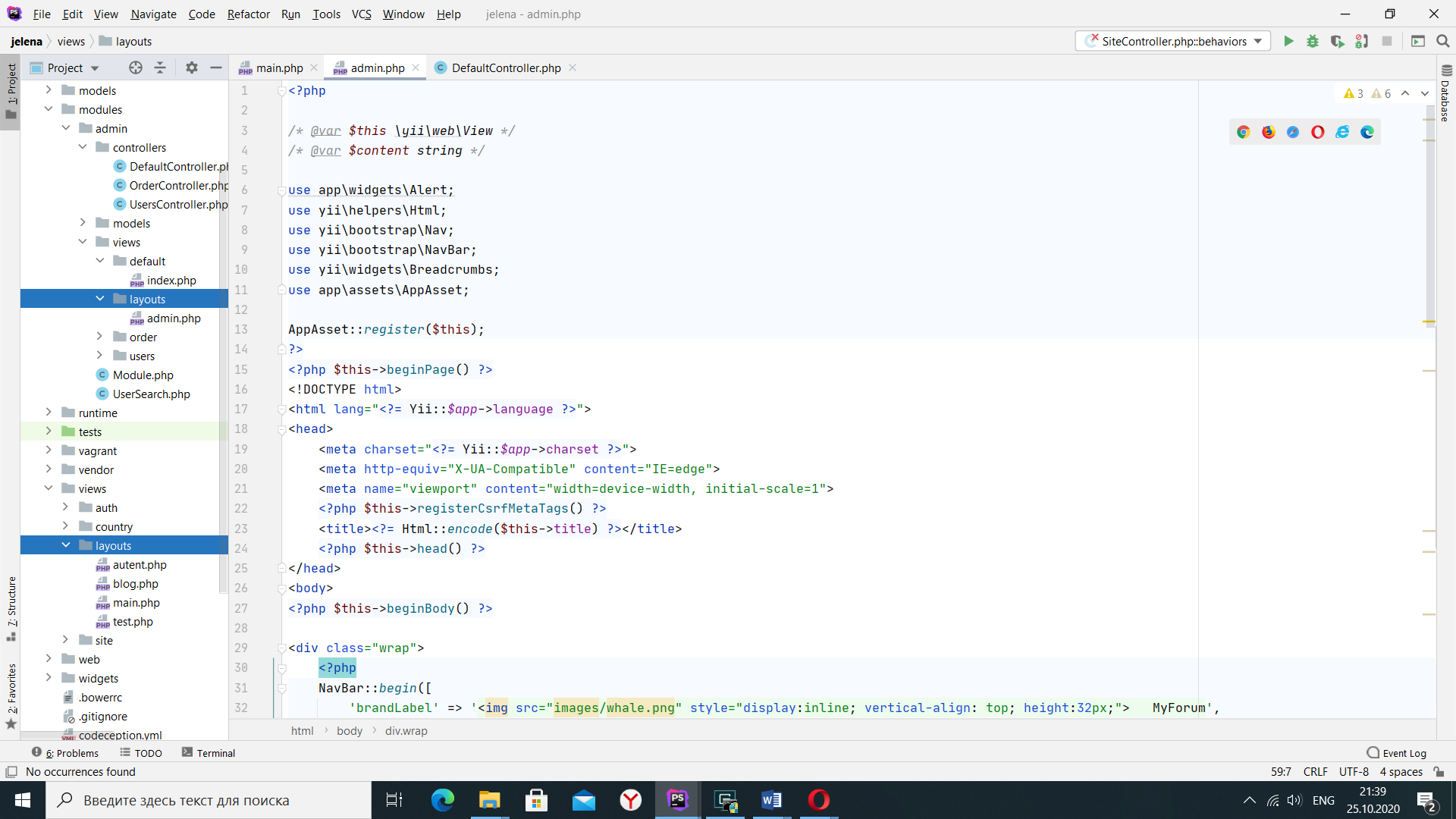
Зайду в папку admin (в папке modules), найду UsersController, подключу там новый шаблон, который назову admin.

Где его создать? Можно в папке layouts папки views, где лежит шаблон main.



Но тогда трудно будет указывать путь к нему, да и это не логично, у нас отдельный модуль для администрирования сайта.

Поэтому создадим внутри папки views (в папке admin) папку layouts, а внутри файл нового шаблона admin.php :

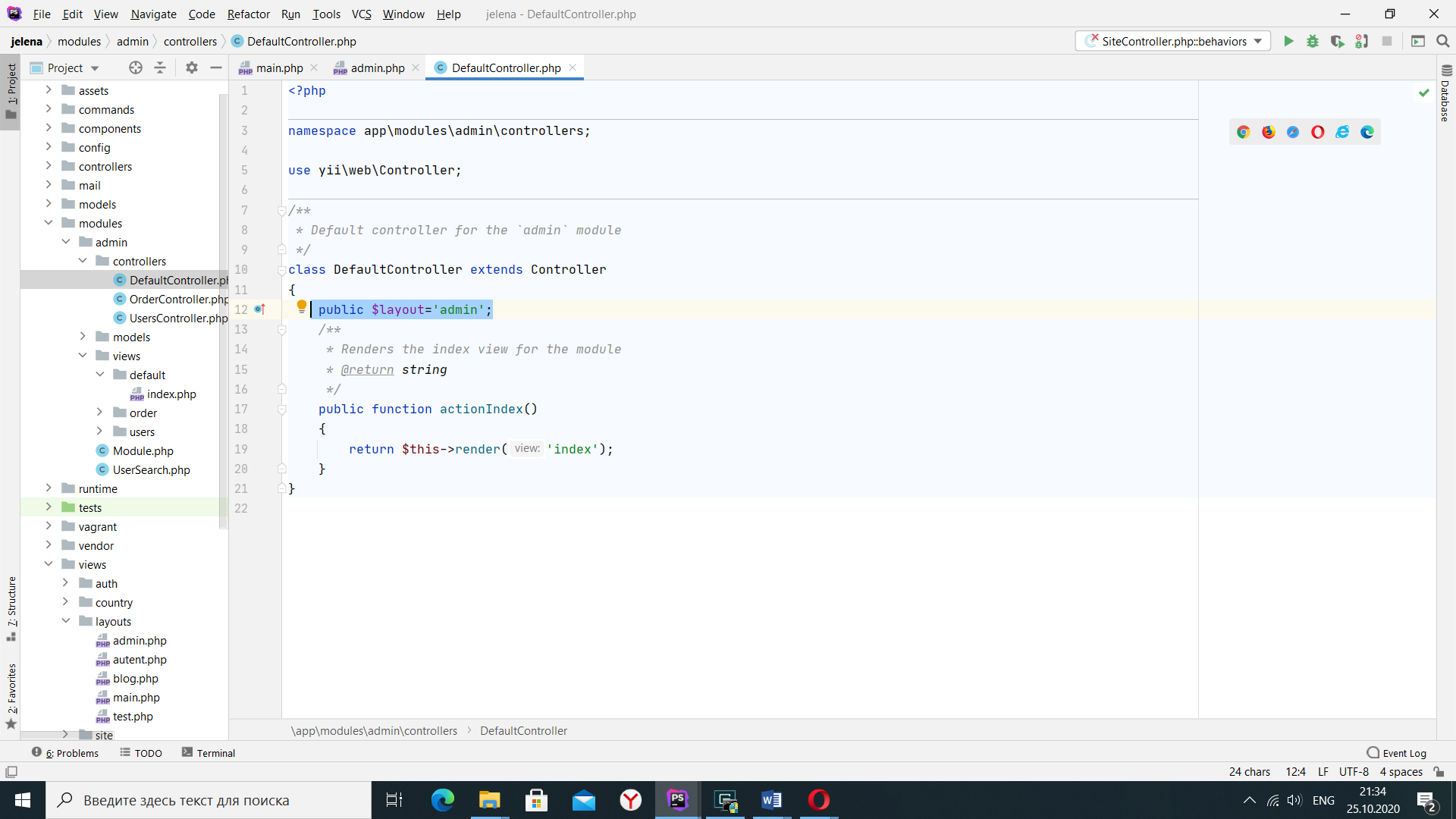


Скопируем из шаблона main код. Там панель, которая нам пригодится.

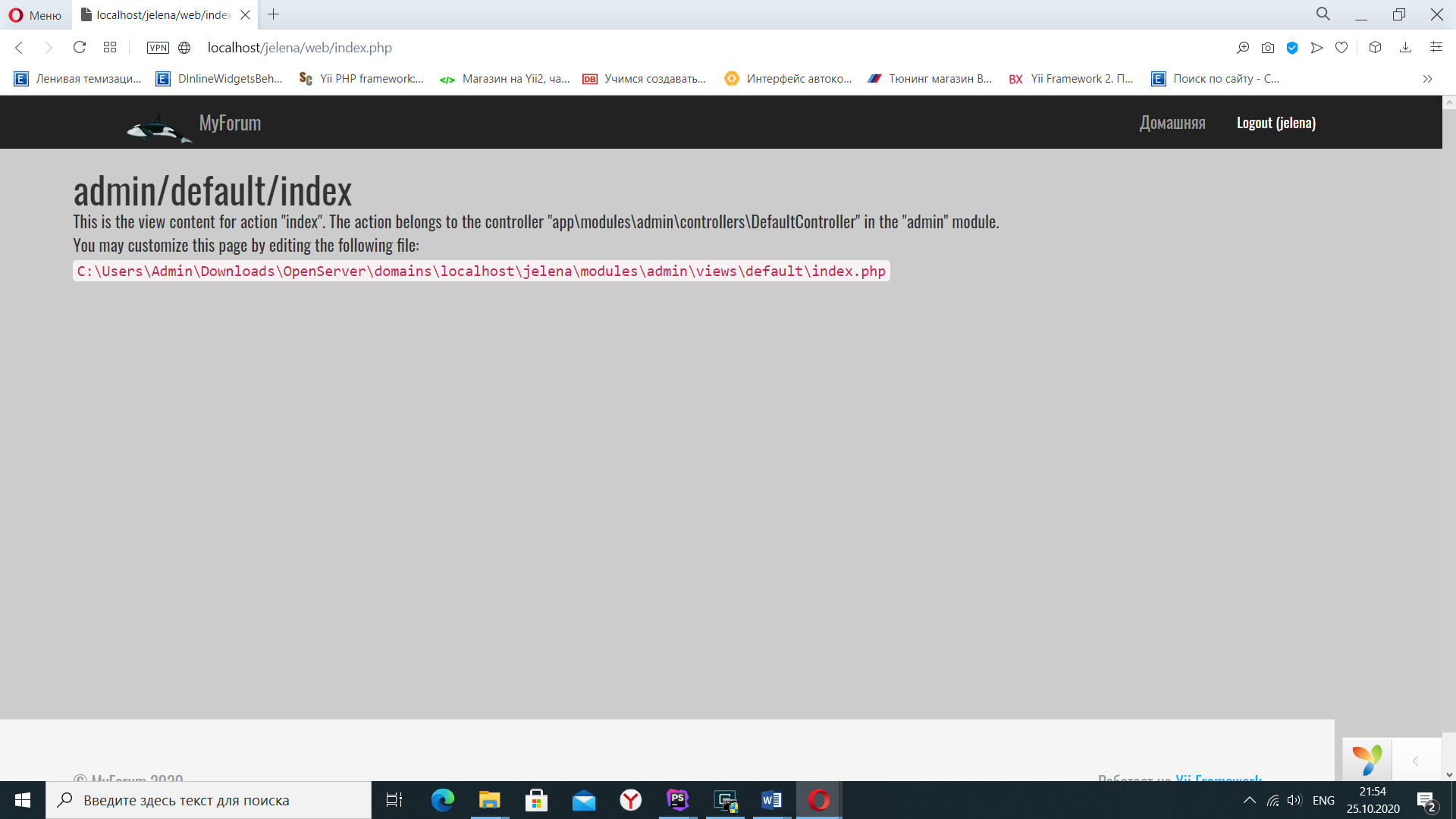
Уберем лишние пункты меню, например, регистрация, авторизация, обо мне, контакт – в админке они не нужны.

Теперь подключим новый шаблон в DefaultController (папка controllers в папке admin):

class DefaultController extends Controller  
{  
 public $layout='admin';

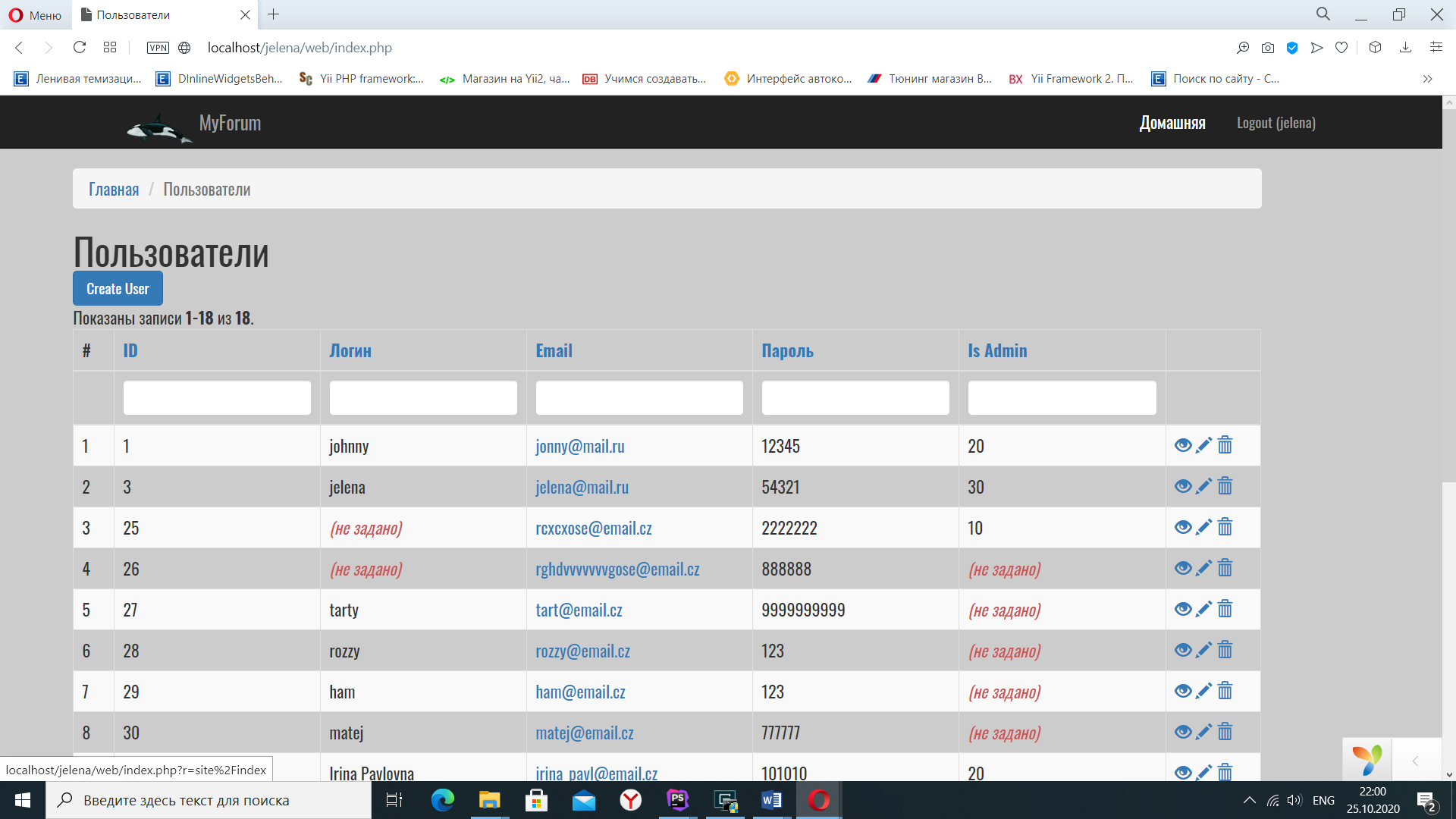


Перейдем в админку, увидим новый шаблон, в котором нет лишних пунктов меню:



Перейдем по ссылке: admin/users/index, увидим, что там лишние пункты в меню остались. Потому что DefaultController отвечает только за файл index.php в админке, значит, надо новый шаблон подключить еще и к контроллеру UsersController.

Подключили:



Иногда необходимо скрыть какой-нибудь пункт меню от пользователя. Например, если пользователь авторизовался, глупо, чтобы он продолжал видеть такие пункты, как «зарегистрироваться» и «авторизоваться».

Вот пример, как это можно сделать с помощью свойства 'visible'(видимость):

['label' => 'Задачи', 'url' => ['/post/show'],'visible' => !Yii::*$app*->user->isGuest],  
['label' => 'Вход', 'url' => ['user/security/login'],'visible' => Yii::*$app*->user->isGuest],

['/post/show']-это контроллер/страница - то есть путь.

Запись Yii::*$app*->user->isGuest означает, что пользователь гость, то есть не авторизован,

Запись !Yii::*$app*->user->isGuest]означает, что пользователь не является гостем, то есть авторизовался.

Скройте по аналогии с приведенным выше примером лишние пункты меню для авторизованного пользователя.

Никаких лишних закладок на панели мы больше не видим:

