1. Работа с Yii-online
   1. Сервисы, позволяющие работать с Yii online

PHP Fiddle – онлайн-редактор для работы с PHP. Можно настроить автозагрузку файлов и подключить Yii, чтобы быстро тестировать код.

Алгоритм:

 Зайдите на [PHP Fiddle](https://phpfiddle.org/).

 В основной области кода можно вставлять PHP-код, связанный с Yii, например, работу с компонентами или небольшие примеры методов Yii.

 Нажмите на кнопку **Run**, чтобы выполнить код.

 Для интеграции Yii-функционала придется загружать его вручную, что не всегда возможно в онлайн-редакторах.

* 1. Gitpod – облачная среда для разработки, поддерживающая различные фреймворки и языки. Можно настроить репозиторий с Yii и работать прямо из браузера.

Алгоритм:

 Создайте аккаунт на [Gitpod](https://www.gitpod.io/).

 Найди или создай репозиторий с проектом на Yii (например, на GitHub).

 Откройте репозиторий через Gitpod. Если репозиторий уже есть на GitHub, просто добавьте в URL префикс gitpod.io/#, например:

|  |
| --- |
| https://gitpod.io/#https://github.com/username/project |

 В среде **Gitpod** установится готовая виртуальная машина для работы с проектом.

 Запустите команду для установки Yii:

|  |
| --- |
| composer install |

|  |
| --- |
| composer create-project yiisoft/yii2-app-basic my-yii-app |

* create-project — команда Composer для создания нового проекта.
* --prefer-dist — флаг, который говорит Composer использовать готовые дистрибутивы (архивы) для установки (это быстрее, чем клонировать репозиторий).
* yiisoft/yii2-app-basic — базовый шаблон проекта Yii 2.
* my-yii-app — имя папки, в которую будет установлен проект. Ты можешь выбрать любое имя.

После установки можно работать с Yii как с локальным проектом, используя консольные команды, настраивая базу данных и другие компоненты.

* 1. Codeanywhere – облачная IDE, которая позволяет работать с любыми проектами на PHP. Здесь можно развернуть Yii-проект и работать с ним через веб-интерфейс.

Алгоритм:

 Зарегистрируйся на [Codeanywhere](https://codeanywhere.com/).

 Создай новое "Container" окружение, выбрав образ с **PHP**.

 В созданной среде подключи проект с Yii. Если начинаешь с нуля, установи Yii командой:

|  |
| --- |
| composer create-project yiisoft/yii2-app-basic my-yii-app |

 Открой проект и начни работать с его файлами, используя онлайн-редактор Codeanywhere.

 Используй терминал для запуска Yii-команд:

|  |
| --- |
| php yii serve |

После этого можно открывать свой проект через предоставленный URL-адрес и работать с ним в браузере.

* 1. Paiza.io – поддерживает работу с PHP и другими языками программирования. Хотя Yii не установлен по умолчанию, его можно настроить для работы с ним.

Алгоритм:

 Перейди на Paiza.io и зарегистрируйся.

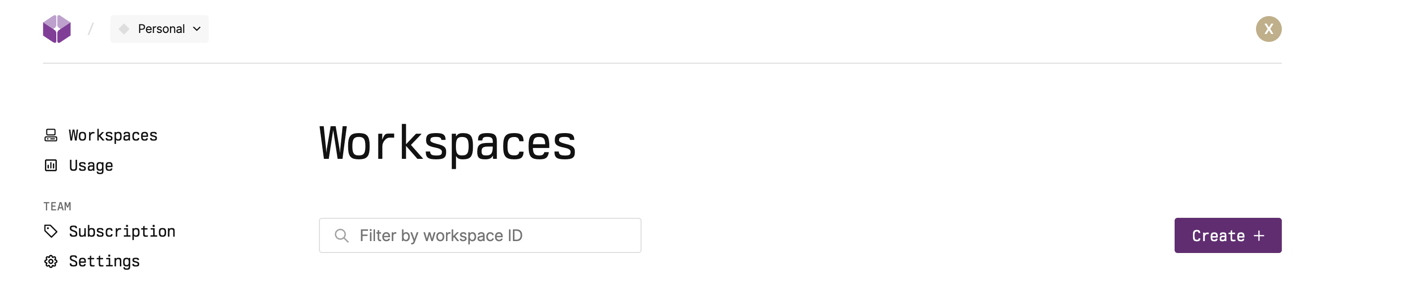
 Создай новый проект, выбрав **PHP**.

 В консоли выполни команду установки Yii через Composer:

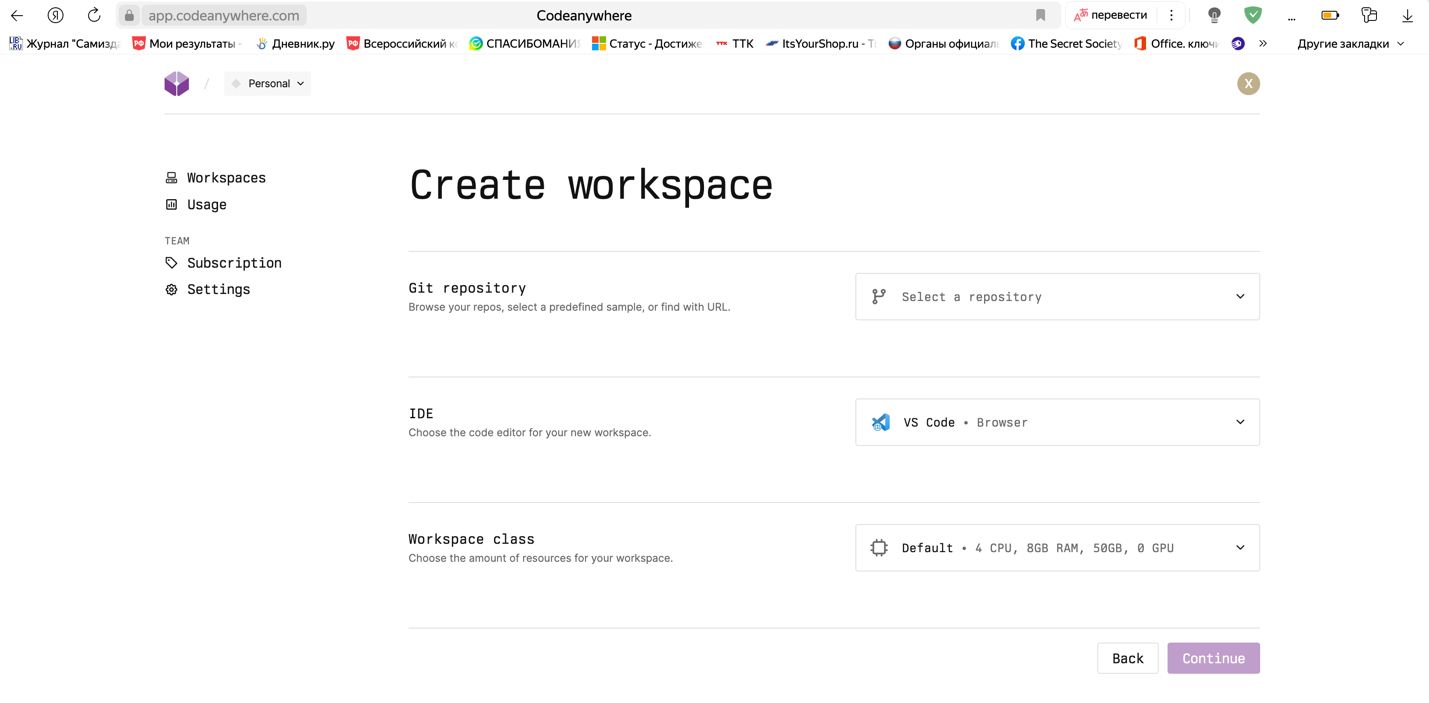
|  |
| --- |
| composer create-project yiisoft/yii2-app-basic my-yii-app |

Работа через Codeanywhere:

После того, как вы зарегистрировались на онлайн-сервисе (регистрация возможна и через сторонние аккаунты, в моем случае, зашла через свой ГитХаб) появляется возможность создание своего рабочего пространства на удаленном сервере.



Как только вы в личном кабинете нажали кнопку create у вас появится окно настроек:



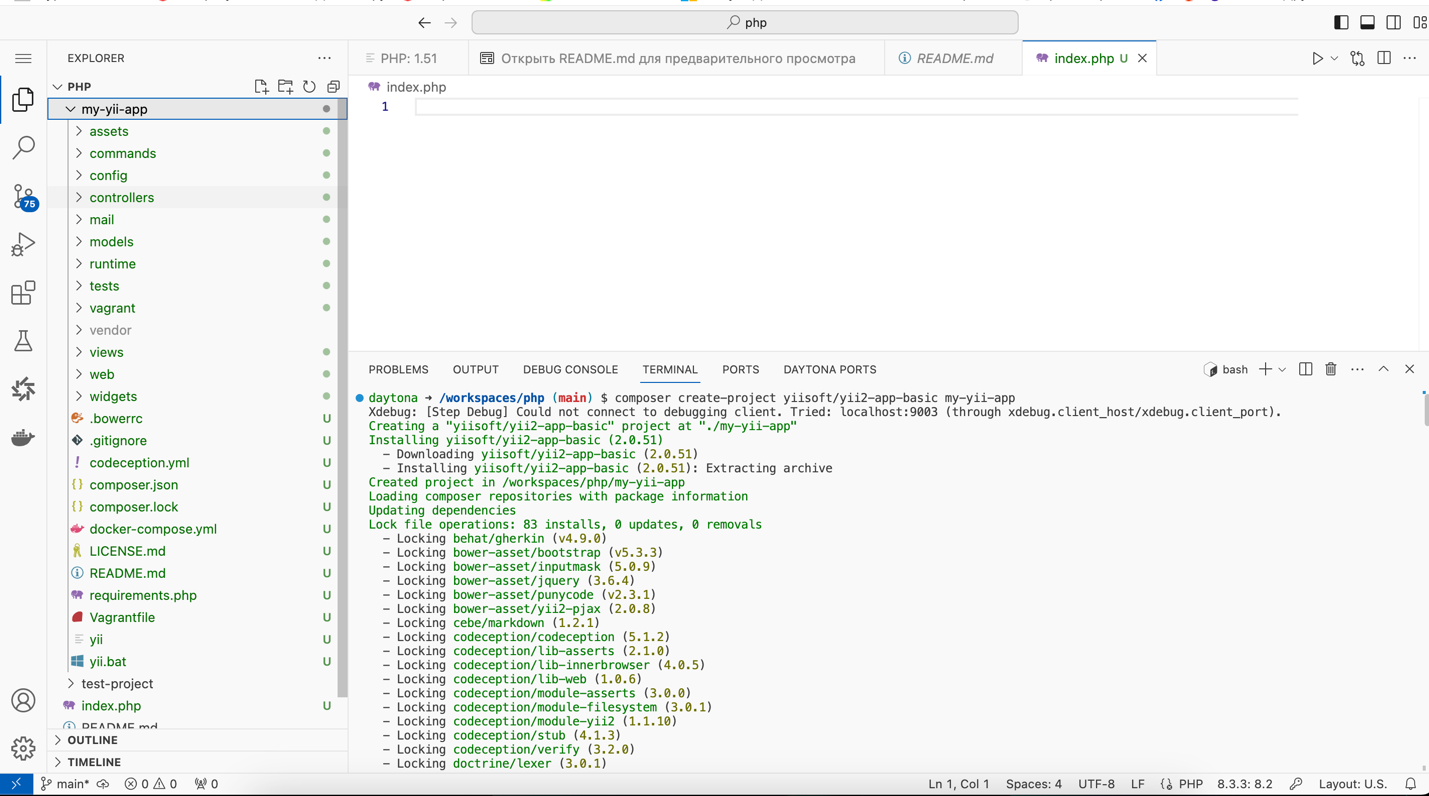
Фактически достаточно выбрать Git repository – выбираете тот ЯП, который будете использовать и можно создавать проект.

Для работы именно с Yii, необходимо открыть терминал:



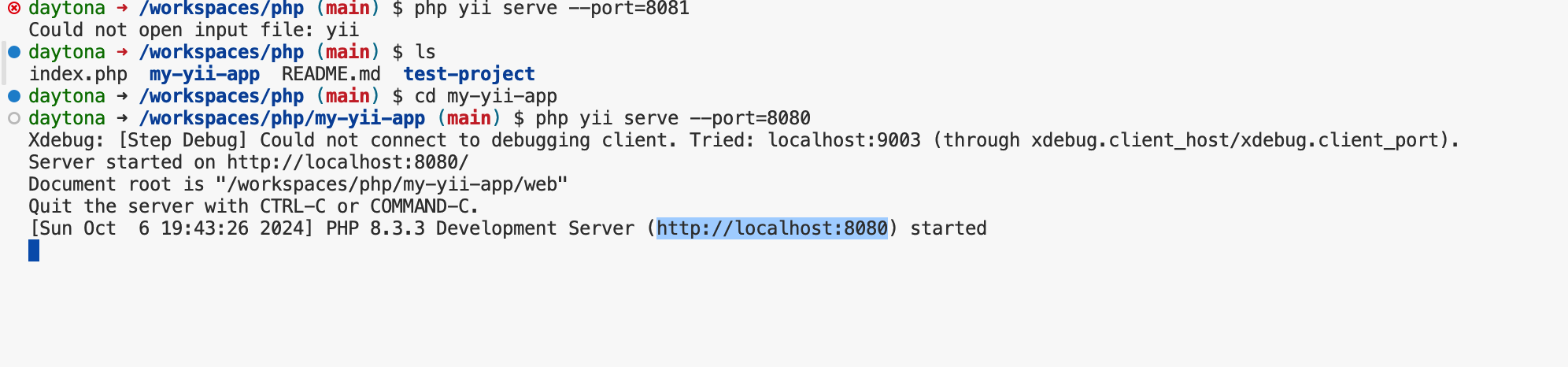
В терминале необходимо будет установить Yii:

|  |
| --- |
| composer create-project yiisoft/yii2-app-basic my-yii-app |



И запустить сервер Yii

|  |
| --- |
| php yii serve --port=8080 |



При запуске сервера обязательно следует убедиться, что вы находитесь в нужной директории (папке вашего проекта). Проверить данные, хранящиеся на вашем сервере можно командой ls и при необходимости, следует перейти в папку вашего проекта (в этом случае, my-yii-app)

URL-адрес сервера будет располагаться в логах терминала.

Чтобы открыть URL-адрес самого проекта необходимо будет использовать публичный URL, который предоставила платформа. Тогда адрес сервера будет выглядеть следующим образом:

|  |
| --- |
| https://<preview-url>:8080/ |

Если что-то не отображается, то можно проверить ошибки, посмотрев логи в файлах:

проверить логи в файле runtime/logs/app.log

1. Работа с Yii локально

* **Запусти VS Code**.
* Выбери **File** (Файл) в верхнем меню.
* Нажми на **Open Folder** (Открыть папку).
* В открывшемся окне выбери папку, в которой находится твой проект Yii (например, my-yii-app), и нажми **Select Folder** (Выбрать папку).

ИЛИ

|  |  |
| --- | --- |
| cd path/to/my-yii-app | cd path/to/my-yii-app |
| code . | code . |

*\*Не забываем, что предварительно проект нужно создать!!!!*

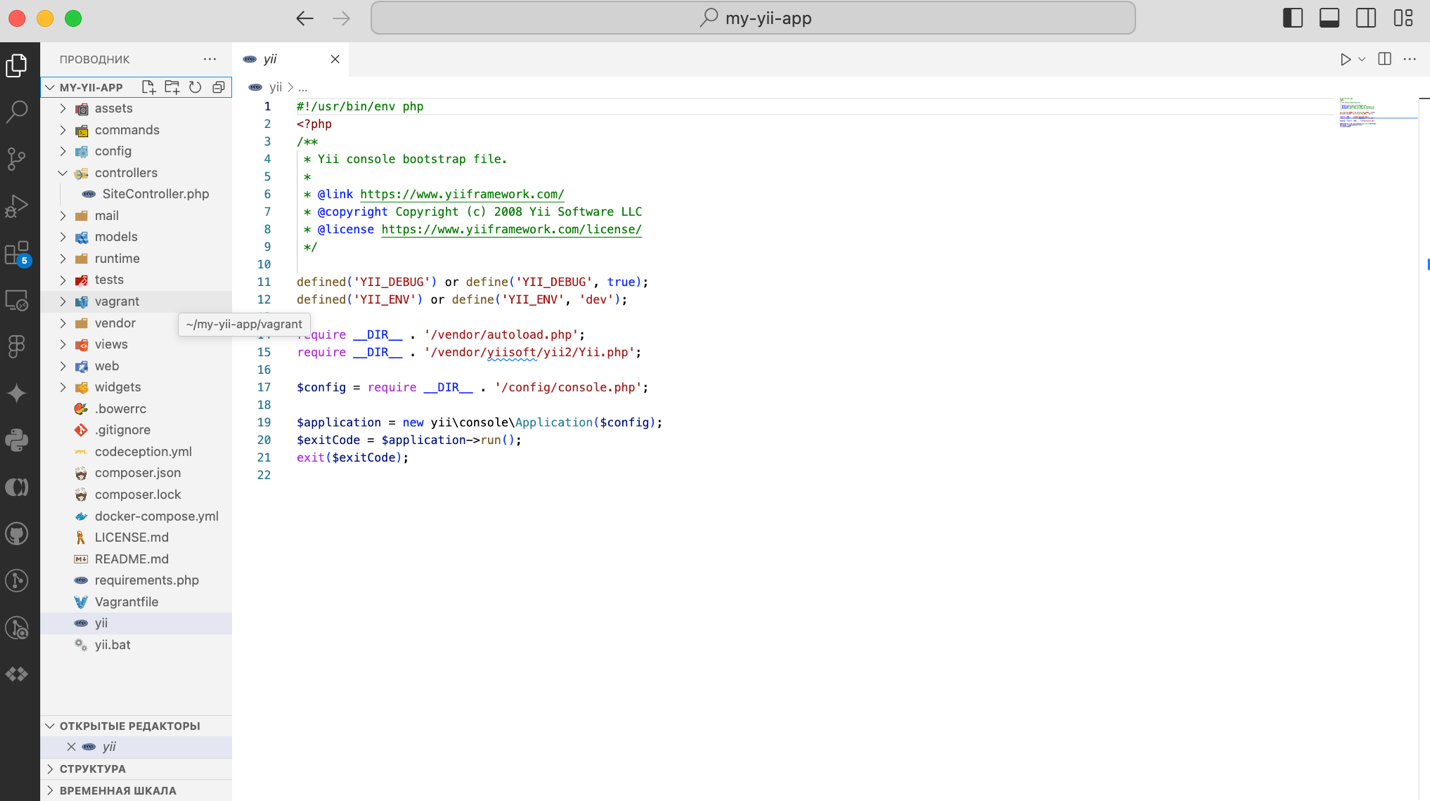
*Установка расширение на VSC (по желанию):*

** ***PHP Intelephense****: для автодополнения и поддержки PHP.*

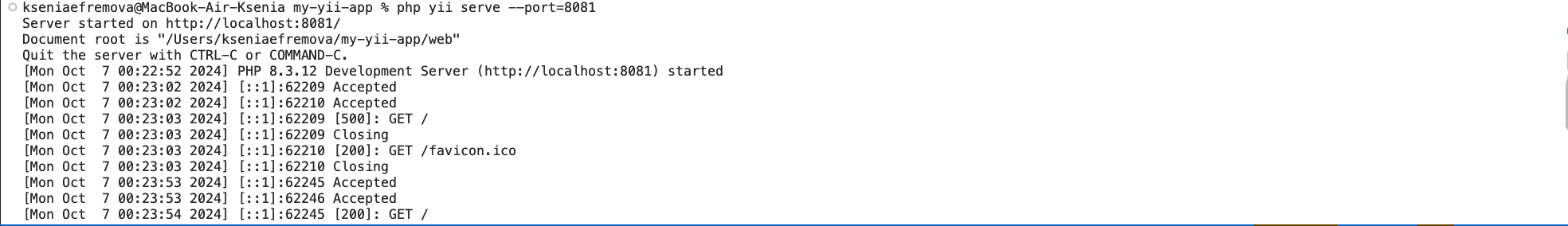
** ***PHP Debug****: для отладки PHP-кода.*

** ***Yii2 Snippets****: для удобного использования сниппетов кода Yii.*

При открытии нужной папки вы увидите структуру Yii проекта

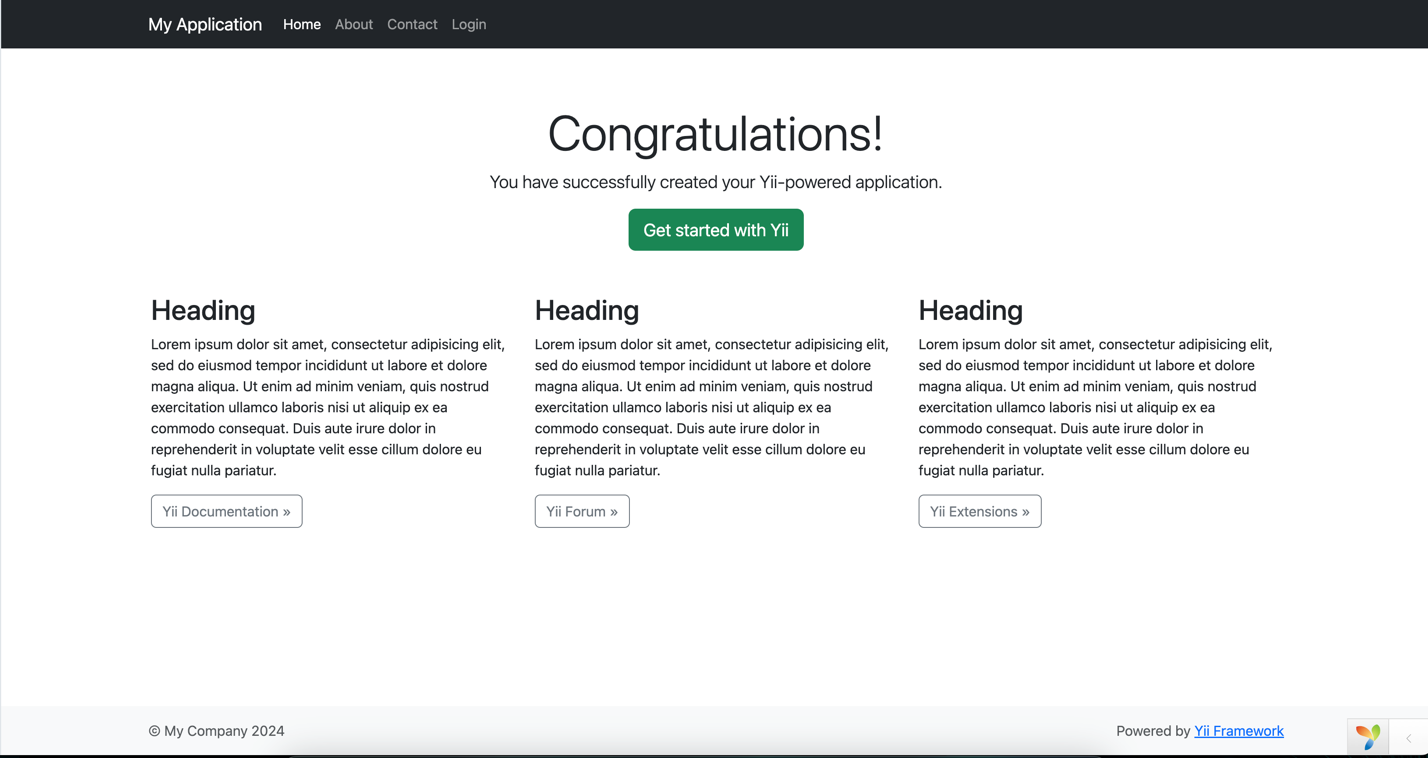


Чтобы просмотреть проект нужно запустить сервер (php yii serve --port=8080)



На этом скриншоте использована команда php yii serve --port=8081, тк порт

8080 был занят.



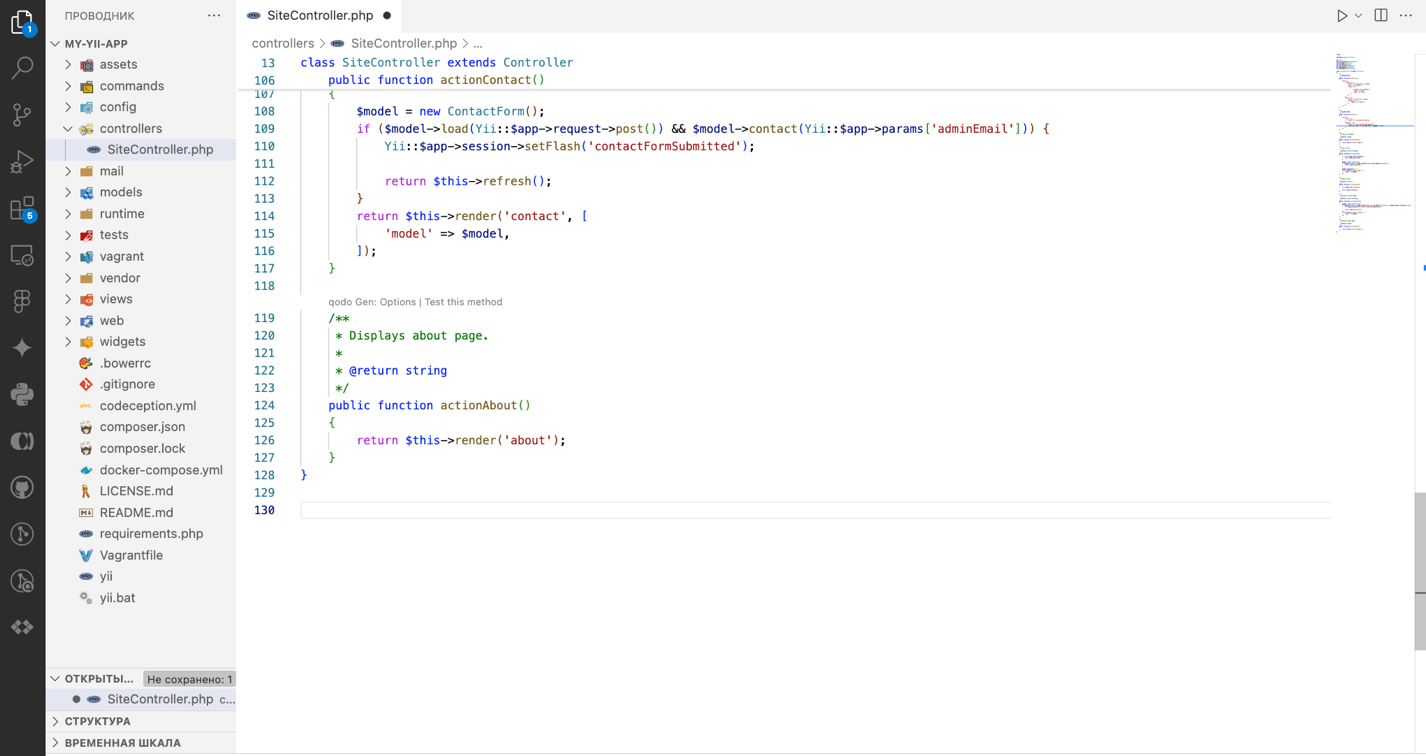
Так выглядит начальная страница экрана только что созданного проекта.

1. Создание простой страницы сайта

Немного модифицируем предложенный код для создания простой страницы сайта.

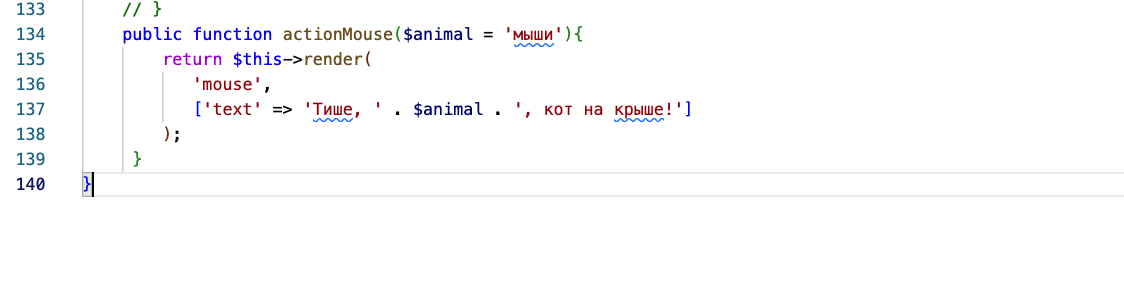
Будем работать с controllers и views

В controllers необходимо открыть SiteControllers и слегка изменить код, переписав функцию actionAbout (находится внизу страницы) на actionMouse

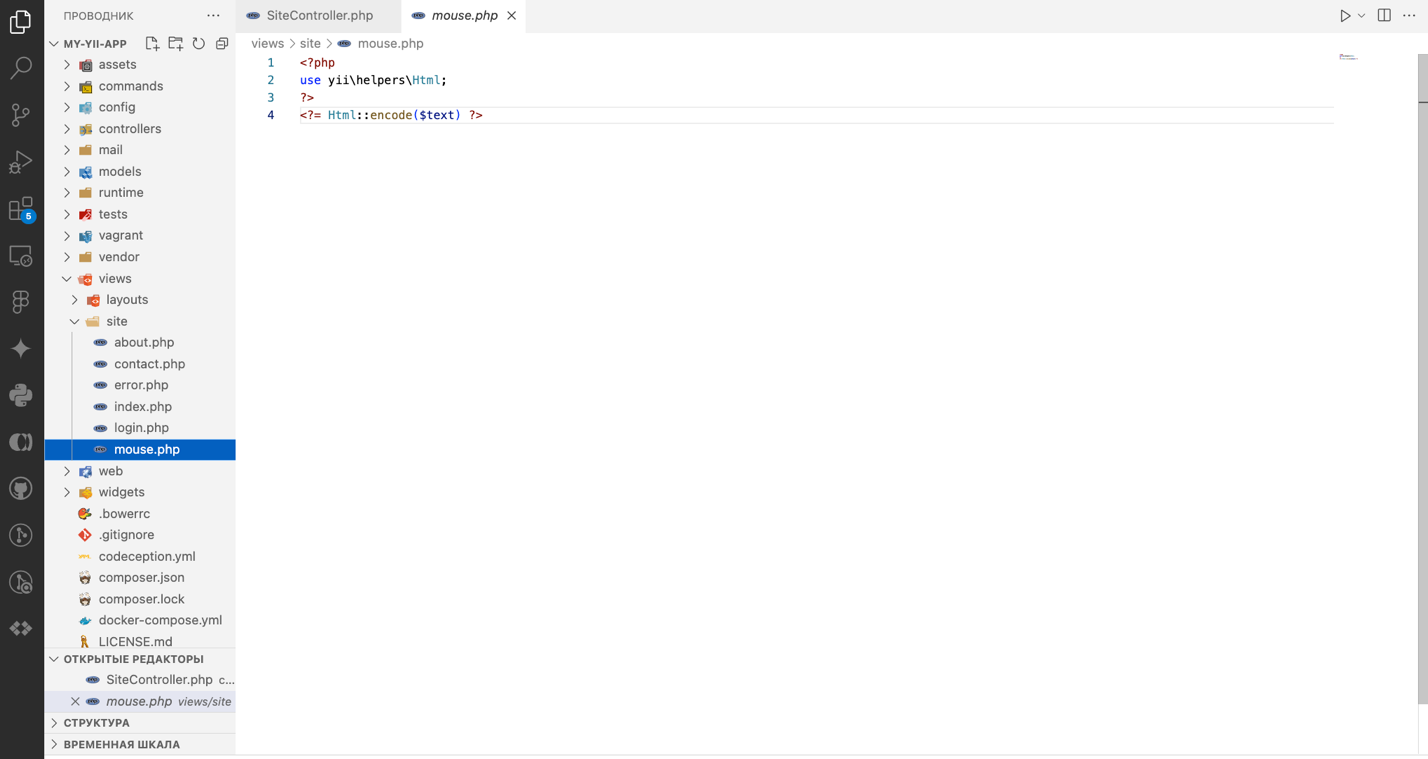


Приставка action в Yii отвечает за распознавание методов действий. В метод "render" передаётся название представления и массив с параметрами. Результат отображения передаётся через "return" в приложение, а затем пользователю.

При вызове метода "render" из действия, он начинает искать PHP файл по адресу views/ID\_контроллера/имя\_представления.php. В нашем случае контроллер называется "Site" (первая часть называния из "SiteController"), а представление называется "mouse" (вторая часть в нижнем регистре из "actionMouse")



Следующий шаг - создание страницы, которая будет выводится при запросе. В папке views необходимо создать файл mouse.php

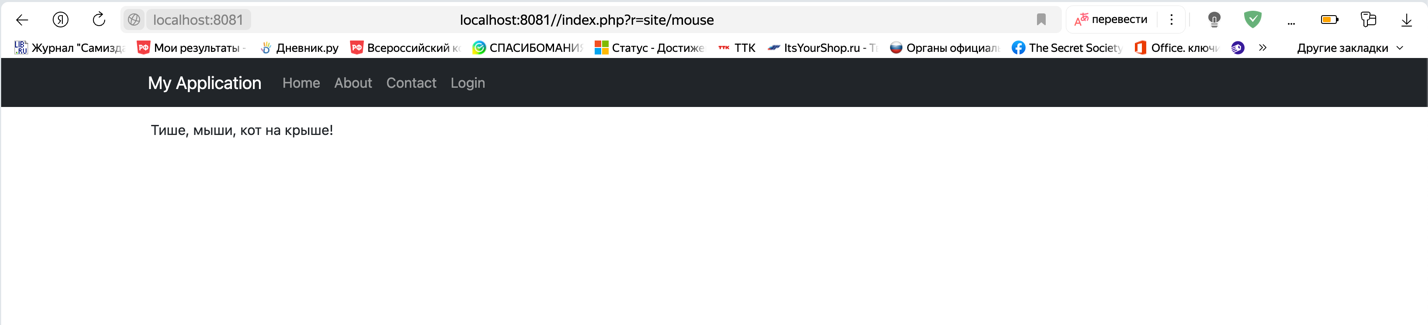


Теперь попробуем запустить страницу. Для этого надо будет дописать путь к созданному файлу:

|  |
| --- |
| /index.php?r=site/mouse |

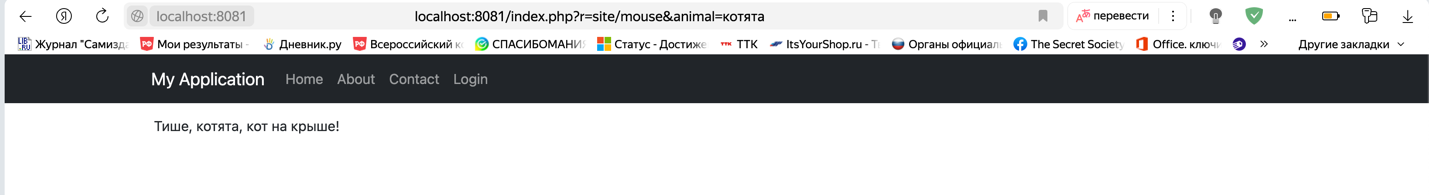
Параметр "r" в строке адреса зарезервирован в Yii для указания контроллера, который обрабатывает запрос.

Обратите внимание на URL- адрес



Метод "actionMouse" может принимать параметр "$animal". предполагается, что этот параметр может быть передан пользователем. Испытаем вывод пользовательских данных, полученных через URL строку параметром. Для этого обратимся к странице:

|  |
| --- |
| /index.php?r=site/mouse&animal=котята |



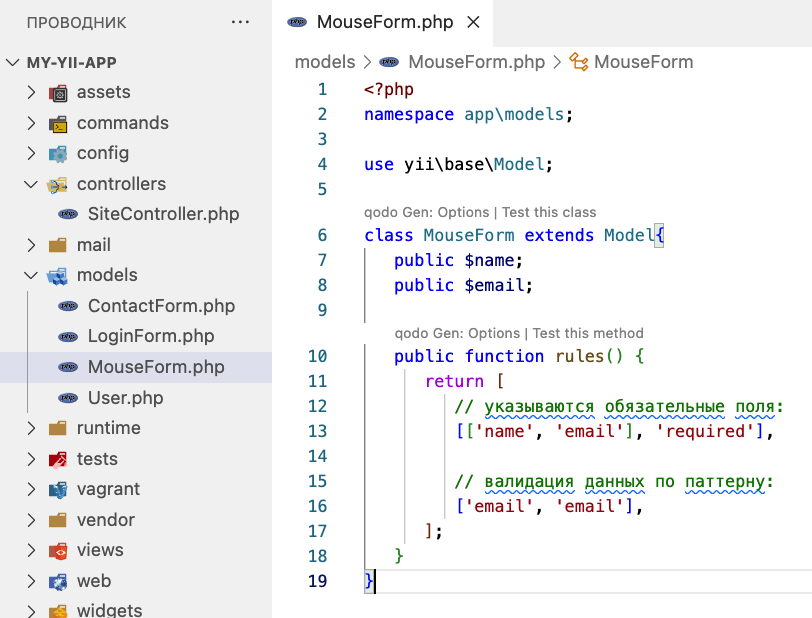
Это пример получения данных приложения через параметр URL.

**Задание 1.**

**Придумайте и примените свой параметр.**

1. Работа с models

Попробуем поработать над простой формой. Сначала создадим файл, в котором будут обрабатываться данные, введённые в форму.



Класс расширяет стандартную "Model" Yii, которая сделана как раз для работы с данными форм. В единственном метода "rule" принимаемые данные имени и мэйла пользователя проходят проверку на обязательность заполнения, а почтовый адрес ещё проверяется на валидность.

Для формы понадобится 2 страница: одна отвечает за вывод на экран формы, а другая за то, куда попадет пользователь после заполнения данных. При создании формы стоит учитывать, что существует:

**field()** — это метод в Yii2 (относится к компоненту ActiveForm) , который помогает автоматически создавать и связывать поля формы с моделью, а также задавать их валидацию.

Оператор -> в PHP используется для доступа к свойствам и методам объекта. Это означает, что он применяется, когда необходимо работать с экземпляром класса, чтобы вызвать метод или получить доступ к свойству этого экземпляра

Поэтому для создания HTML – элемента формы прописывается конструкция примерно такого вида:

|  |
| --- |
| echo $form->field($model, 'username')->textInput() |

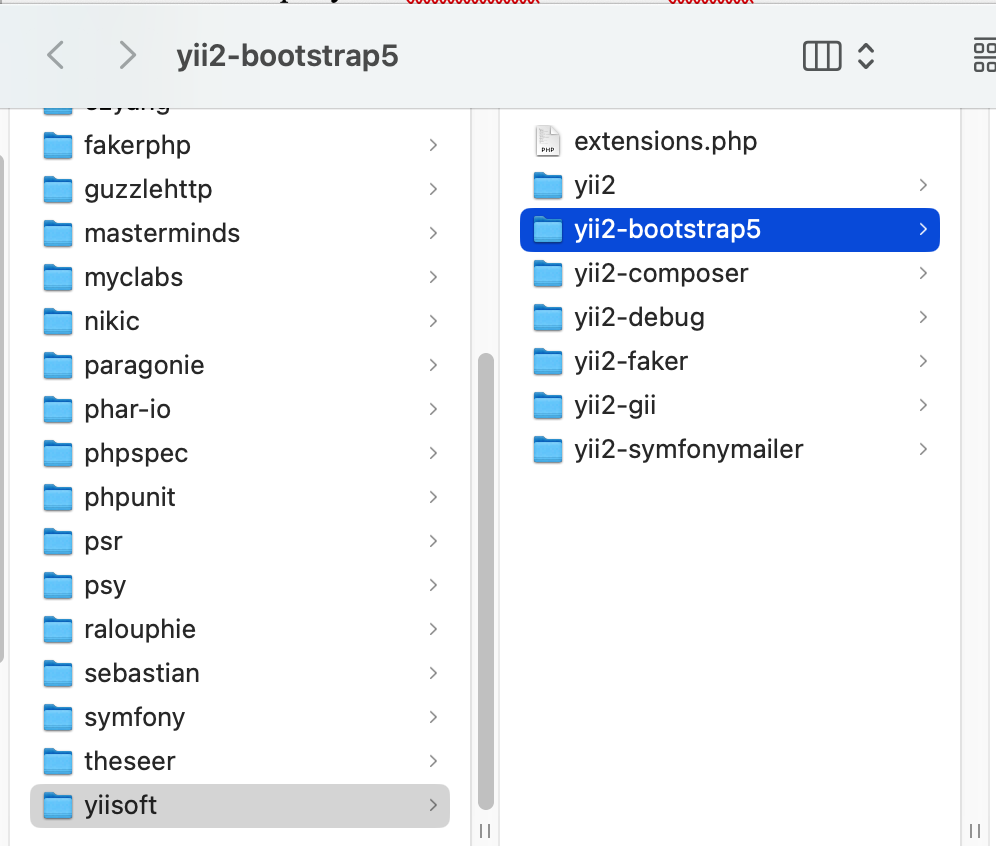
 field($model, 'username'): Это метод, который создает HTML-элемент формы (например, <input>) и связывает его с моделью. В данном случае поле формы связывается с атрибутом username модели $model.

 Модель $model содержит данные, которые вводит пользователь, и валидационные правила для этих данных.

->textInput(): Это метод, который указывает тип HTML-элемента (в данном случае — текстовое поле <input type="text">).

|  |
| --- |
| **$form->field($model, 'email')->input('email')** |

Это создает поле для ввода email с типом email (<input type="email">). Поля с типом email автоматически проверяются браузером на наличие валидного email-адреса.



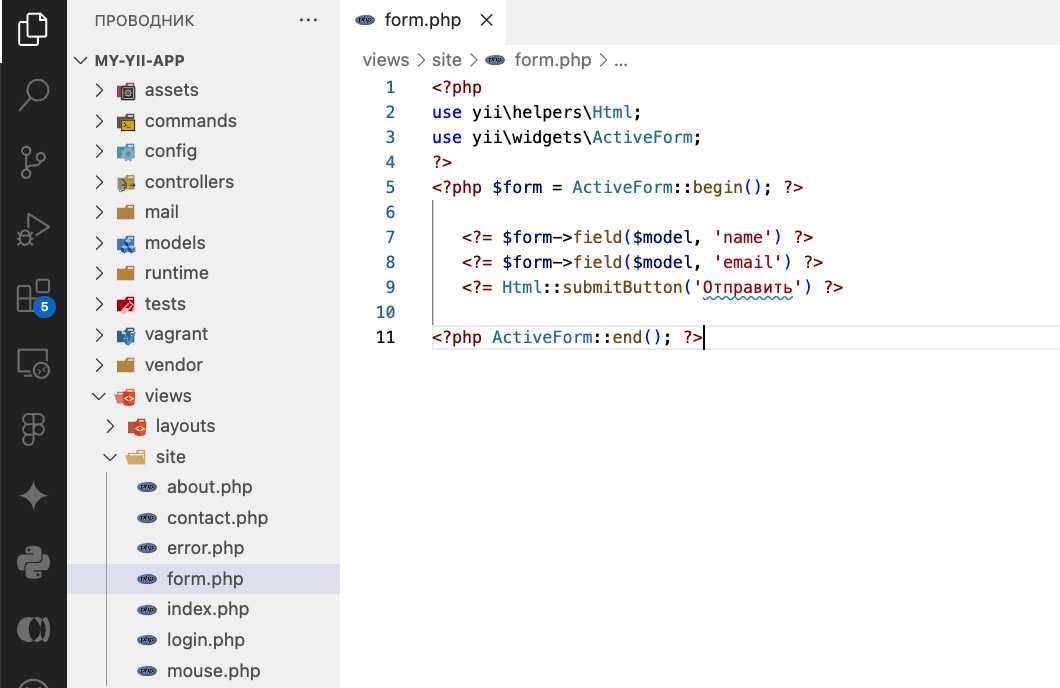
Yii2 включает в себя компоненты Bootstrap в своих стандартных шаблонах. Если вы создаете проект на основе базового приложения (yii2-app-basic) или продвинутого приложения (yii2-app-advanced), Bootstrap уже будет подключен через файлы CSS и JavaScript.

Поэтому можно использовать классы Bootstrap прямо в своих представлениях. Например, для создания кнопок, форм, навигационных панелей и других компонентов интерфейса.

|  |
| --- |
| **Html::submitButton('Отправить', ['class' => 'btn btn-primary'])** |

Создает кнопку отправки формы с текстом "Отправить" и CSS-классом btn btn-primary (класс Bootstrap), чтобы стилизовать ее как кнопку Bootstrap.

Сама форма: (views/site/form.php)



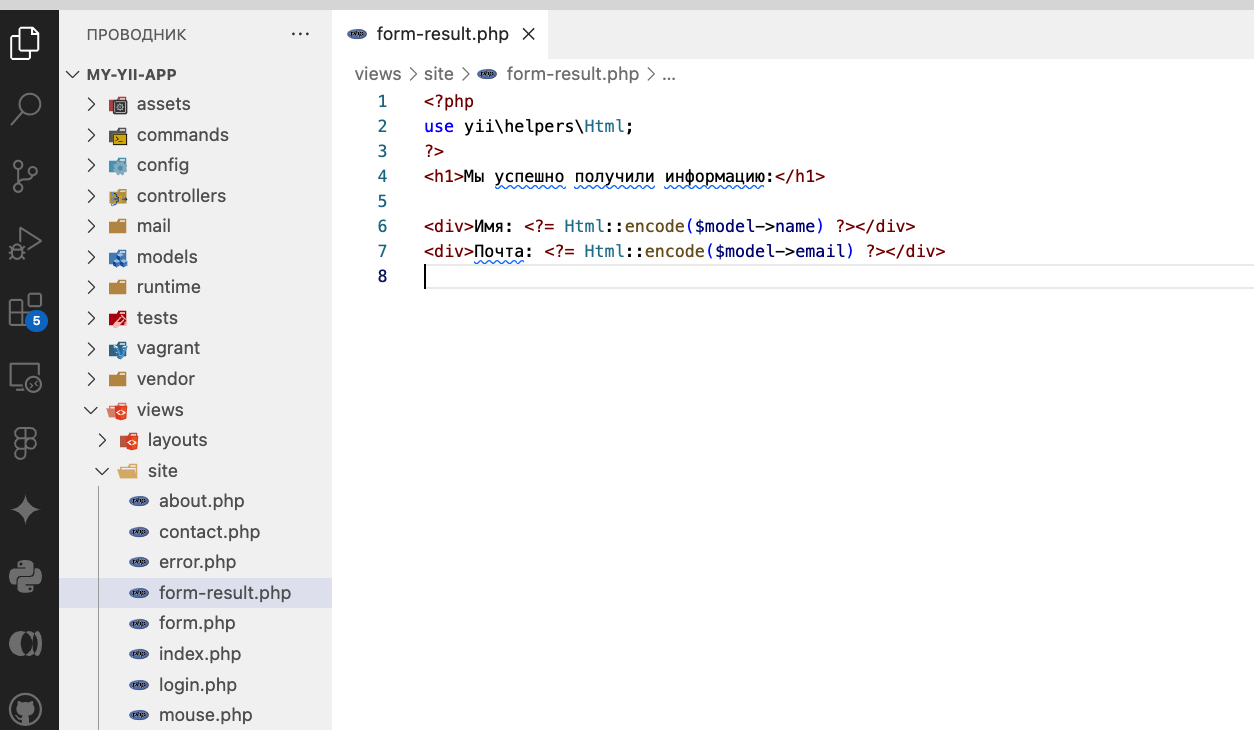
ActiveForm::begin()

Этот метод используется для открытия формы в представлении. Он генерирует HTML-код, необходимый для начала формы и связывает ее с моделью, если она передана в качестве параметра. Генерирует открывающий тег <form>

ActiveForm::end()

Этот метод используется для закрытия формы. Он генерирует закрывающий тег </form> и выполняет некоторые операции очистки.

Файл с информацией об успешной отправки формы: (views/site/form-result.php):



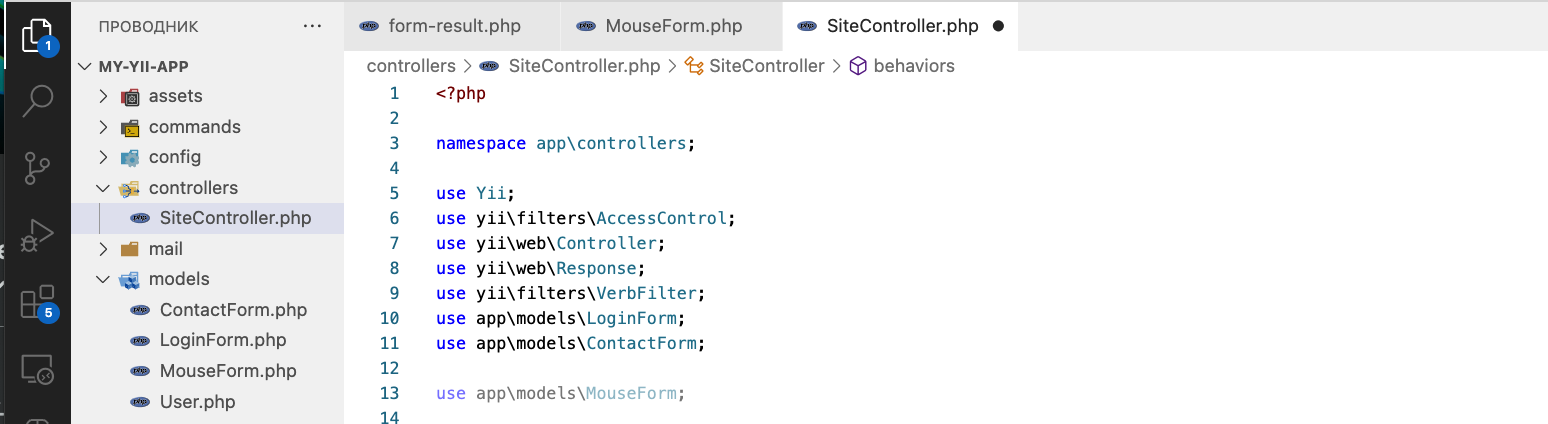
Вся принимаемая от пользователей информация должна быть безопасна. Если требуется вывести её на экран после отправки формы, то следует преобразовать HTML сущности с помощью "Html::encode".

Html::encode()

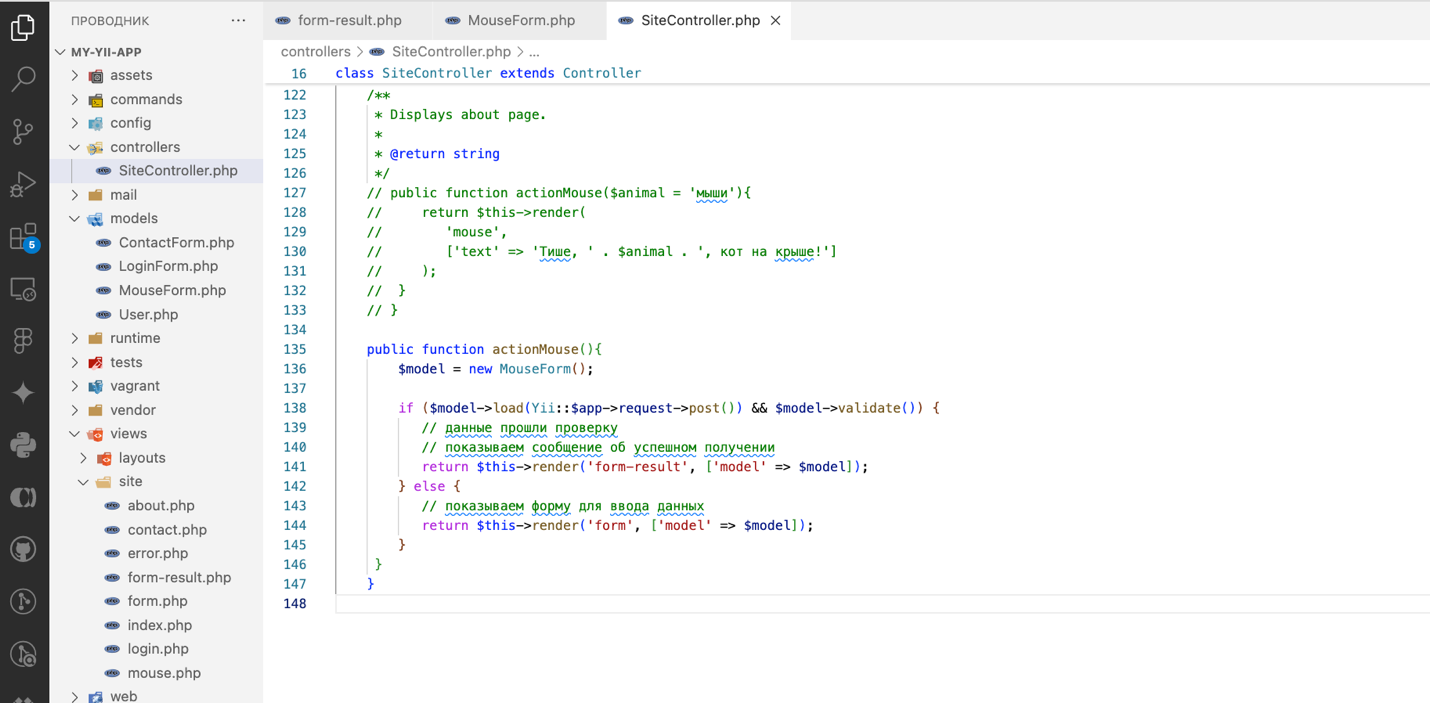
В Yii используется для экранирования (кодирования) специальных HTML-символов в строке. Это важно для предотвращения XSS-атак (межсайтового скриптинга), когда пользователь вводит данные, которые могут содержать вредоносный код. Когда пользователь вводит данные существует риск, что эти данные могут содержать специальные символы HTML (например, <, >, &), которые могут нарушить структуру HTML или, что еще хуже, позволить злоумышленнику выполнить вредоносный код на вашей странице.

Теперь пропишем контроллер для нашей формы:

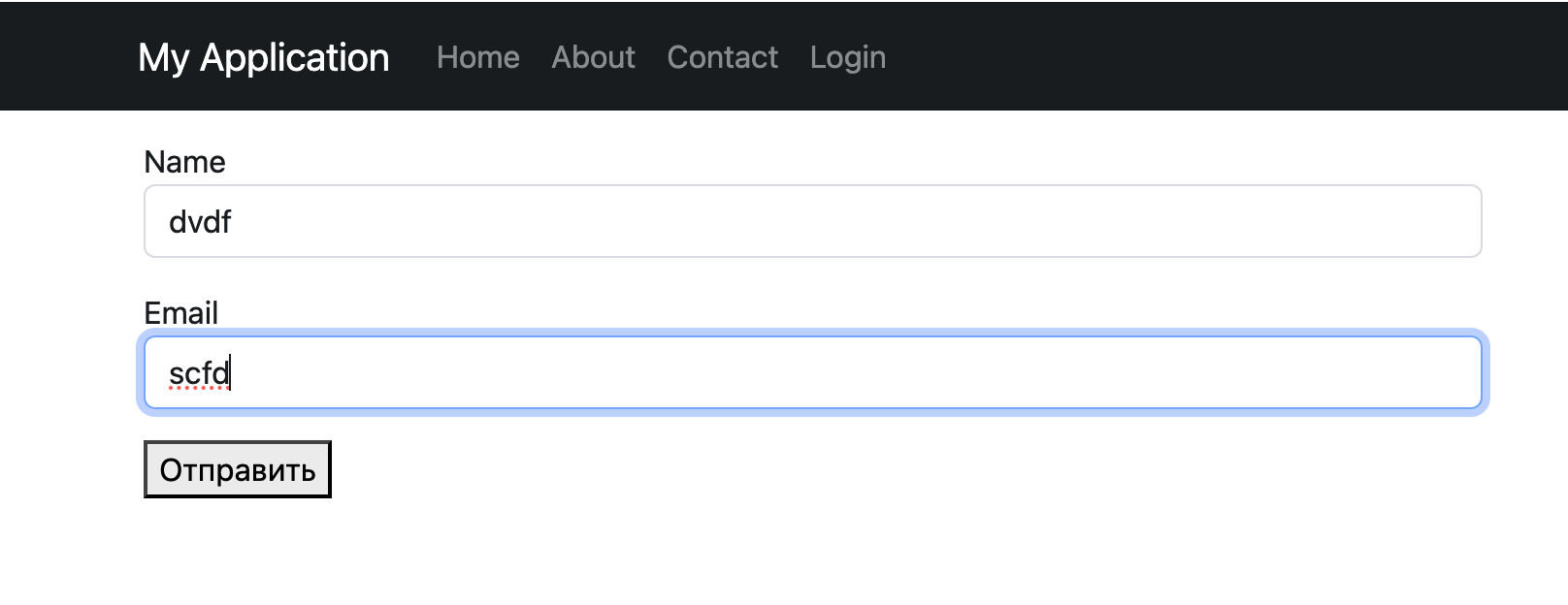
-первое что мы делам это передаем информацию о созданной модели формы в файл SiteController



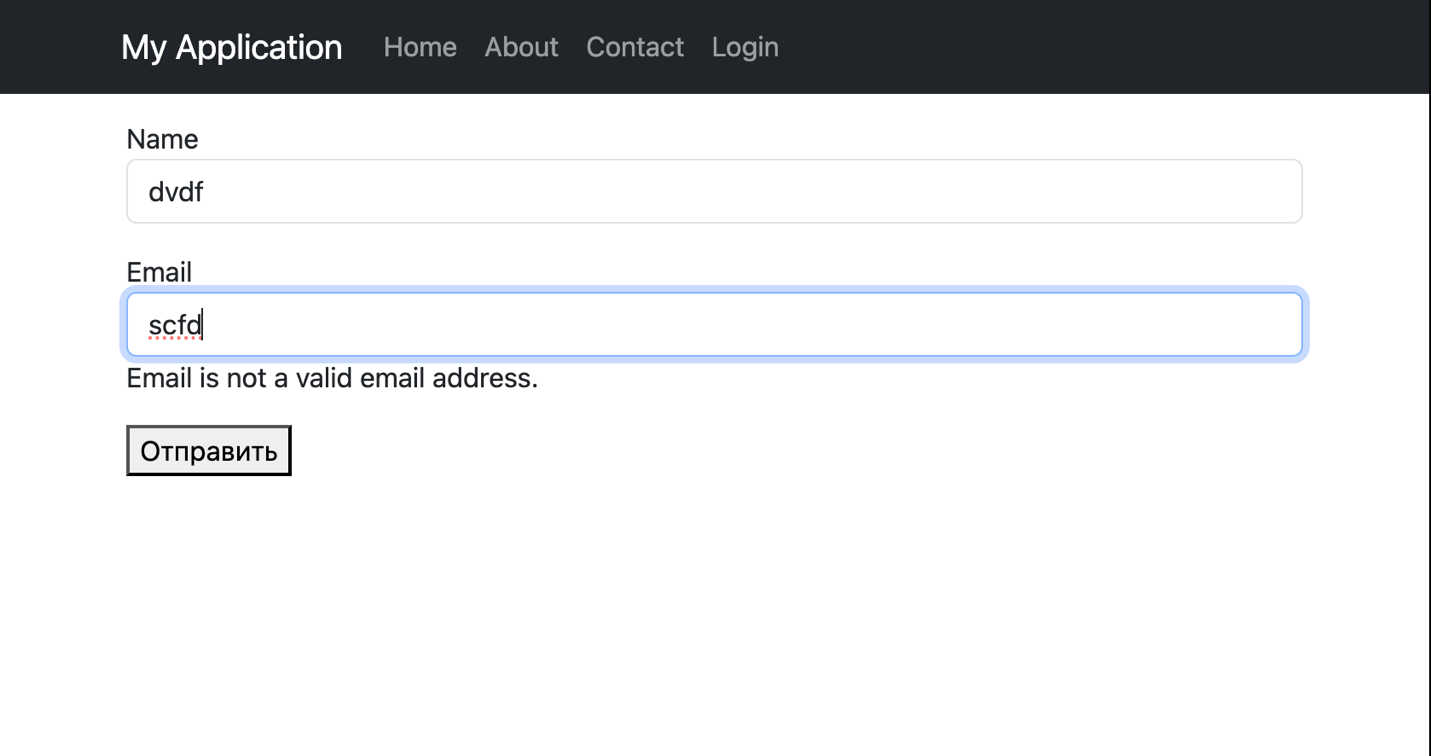
Внизу страницы добавляем функцию-действие (предварительно убрав старую функцию):



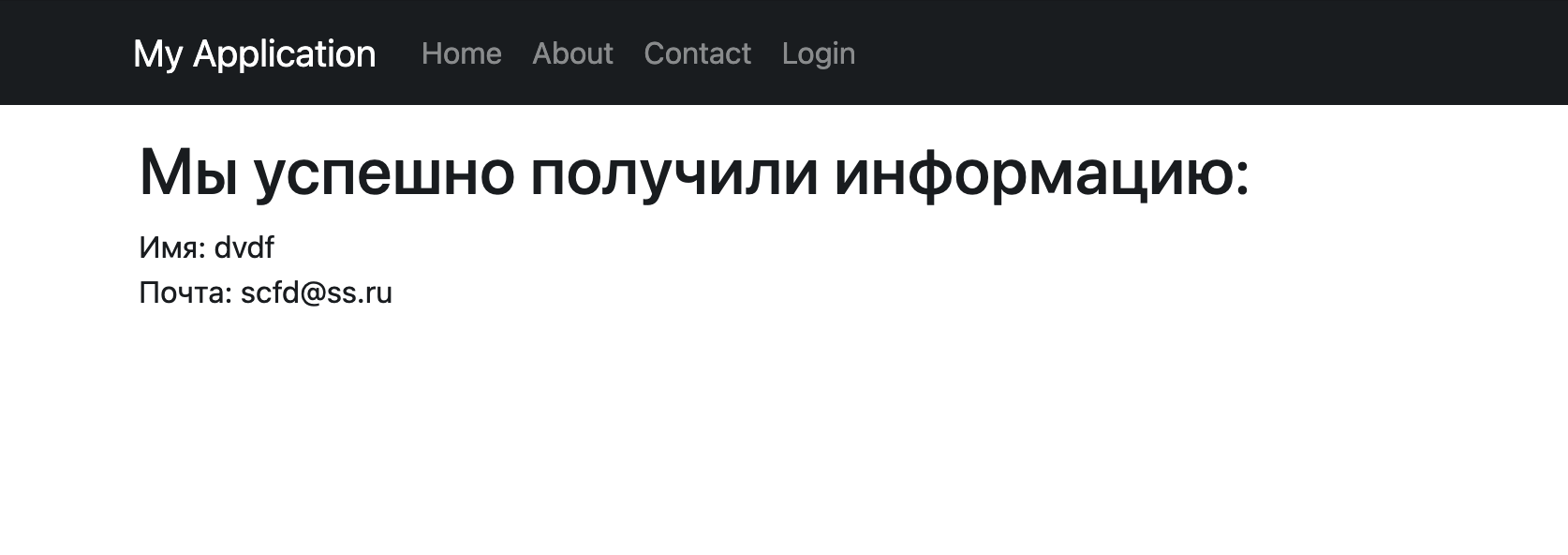
Чтобы проверить верно/неверно отображаются прописанные нами данные, открываем нужную страницу (предварительно добавив /index.php?r=site/mouse). Если вес сделано верно, то на экране отобразится следующая форма:



При неверно заполненном поле будет выводится ошибка:



Ошибка в заполнении email. При исправленной ошибке данные отправляются, а пользователя перебрасывает на другую страницу



**Задача 2. Добавьте дополнительное поле в форму (например, family name)**

**Контрольные вопросы:**

1. **Что такое пространство имен (namespace) в PHP, и как оно используется в Yii? Приведите пример.**
2. **Как создать новую модель в Yii и какие основные методы вы должны реализовать в этой модели?**
3. **Опишите, что делает метод ActiveForm::begin(). Каковы его основные параметры?**
4. **В чем разница между использованием метода Html::encode() и прямым выводом переменных в представлениях? Почему это важно?**
5. **Как вы можете добавить валидацию для поля формы в модели? Приведите пример правила валидации.**
6. **Какова основная архитектура, используемая в Yii, и какие компоненты она включает?**
7. **Опишите структуру директорий, создаваемую после установки Yii. Какие папки в ней присутствуют и за что они отвечают?**
8. **Как создается новый контроллер в Yii, и какой класс обычно используется для этого?**