Титульный лист

МИНИСТЕРСТВО ОБРАЗОВАНИЯ МОСКОВСКОЙ ОБЛАСТИ

Государственное бюджетное профессиональное образовательное учреждение Московской области «Ногинский колледж»

Курсовой проект

по МДК.09.01 Проектирование и разработка веб-приложений

ПМ.09 Проектирование, разработка и оптимизация веб-приложений

Тема:

ИСПОЛЬЗОВАНИЕ ФРЕЙМВОРКА YII2 В РАЗРАБОТКЕ ВЕБ-ПРИЛОЖЕНИЯ «СЛУЖБА ДОСТАВКИ ЕДЫ «СЫРФЫРС»»

Разработчик: студент группы 3ИСПР1

Пронина Е.А.

(подпись)

Оценка защиты курсового проекта

« »

Дата защиты

« » 2024

Руководитель проекта:

преподаватель Степанов С.О.

(подпись)

Ногинск, 2024 г

**ВВЕДЕНИЕ**

В современном мире, с его динамичным развитием технологий и растущей популярностью онлайн-услуг, сфера доставки еды как никогда актуальна. Все больше людей предпочитают заказывать еду на дом или в офис, что привело к появлению множества служб доставки, предлагающих удобные онлайн-платформы и мобильные приложения для заказа еды. В связи с этим, создание эффективных и удобных веб-приложений для служб доставки еды становится актуальной задачей для разработчиков программного обеспечения.

Главной целью данного курсового проекта является разработка функционального веб-приложения для службы доставки еды с использованием фреймворка Yii2, а также его самостоятельное изучение. Yii2 является мощным и популярным инструментом для создания веб-приложений, который предоставляет разработчикам широкий набор функций и возможностей для создания масштабируемых и безопасных приложений.

В ходе написания курсовой работы нам предстоит использовать GitHub для отгрузки проекта на удалённый репозиторий, научиться работать с базой данных MySQL а также загрузить готовое веб-приложение на общедоступный хостинг.

**1. ПРОЕКТНЫЙ ЭТАП**

На этом этапе мы определяем, как должно выглядеть и функционировать наше веб-приложение. В моей голове представляется сайт фастфуд-ресторана с оригинальной авторской едой, сделанной в приоритете из сыра. Можно сразу разделить товары будущего каталога на разделы и поискать картинки, которые бы вписались в дизайн сайта.

Но а перед реализацией своих мыслей в файлах, первым делом создадим удалённый репозиторий на GitHub. Подключим его к папке на рабочем ноутбуке при помощи GitHub Desktop. Эта папка – наш проект, в ней будет фиксироваться каждое изменение файлов.

Начнём с базового шаблона yii2, который мы перенесём в папку при помощи команды в Git Bash:

composer create-project --prefer-dist yiisoft/yii2-app-basic basic

Взглянем на полученные файлы. В папке web хранится лицевая сторона сайта, но в браузере открывается список всего содержимого корневой папки. Создадим в ней файл .htaccess и напишем следующий код:

RewriteEngine on

RewriteRule ^(.+)?$ /web/$1

Теперь при запуске проекта через Open Server Panel нам будет открываться index.php, находящийся в папке web. И можно приступать к созданию своего сайта, который будет отображаться вместо шаблона.

Набросаем небольшой макет в фигме. Здесь определилась общая палитра и стиль веб-приложения, блоки и структура каталога. Теперь можно начать вёрстку шаблона, который вскоре перенесём в бэкенд.

**2. ПЕРЕНОС ШАБЛОНА**

Вёрстка главной страницы уже давно знакомый принцип. Теперь нужно перенести её на главную страницу сайта. Но для начала сделаем layout сайта, его неизменную часть на каждой странице.

У нас есть готовый пример из yii2 шаблона, файл main.php, находящийся в папке views > layouts. Создадим в этой же папке свой файл site.php и перенесём туда из навёрстанного шаблона голову сайта с метаданными, а также тело сайта с хэдером и футером. Остальное же содержание страницы перенесём в index.php, находящийся в папке site, которая и содержит все страницы веб-приложения, регулируемые через SiteController.

При переносе layout нам нужно добавлять особые маркеры, чтобы фреймворк понимал, где начинаются и заканчиваются определённые теги. Без этих маркеров не будут работать стили.

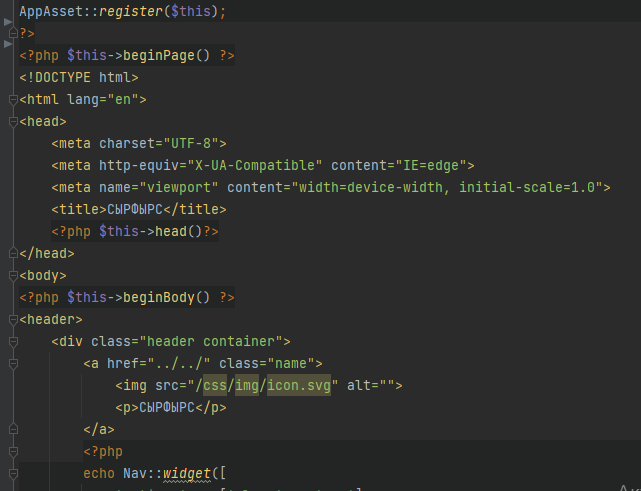


Рис. 2.1. Внедрение маркеров в layout

Файлы со стилем и скриптами мы перенесём в папку web и пропишем их в AppAssets. Через него же мы зарегистрируем layout.

**3. РАБОТА С БАЗОЙ ДАННЫХ**

**3.1. Подключение**

Для работы с базой данных (далее – БД) будем использовать PhpMyAdmin, встроенный в Open Server Panel, пропишем подключение к локальному хосту в config, как на скриншоте ниже.

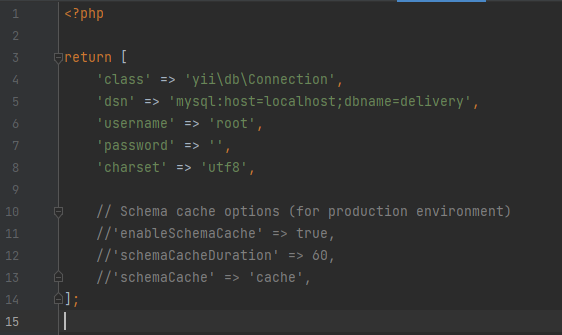


Рис. 3.1.1. Подключение БД

Для создания БД будем использовать миграции.

Миграция базы данных — это функция в Yii, которая позволяет отслеживать изменения структуры базы данных. Миграции могут повлиять не только на схему базы данных, но и скорректировать существующие данные в соответствии с новой схемой, создать иерархию RBAC или очистить кэш.

**3.2. Регистрация и логин**

В шапке сайта первым делом стоит сделать функционал кнопке входа и регистрации. Я написала его по инструкциям из гайдов по фреймворку. Для этого необходимо было создать два представления с формой для регистрации и логина соответственно, три модели: User, LoginForm и SignupForm, а также миграцию для создания БД с пользователями.



Рис. 3.2.1. Создание таблицы user в миграции

В моделях форм мы прописали генерирование ключа аутентифкации и токена для пароля, поэтому сам пароль в БД найти невозможно.

Также мы создали строку статус, которая пригодится позже, когда мы будем делать из пользователей админов и курьеров.

**3.3. Создание каталога**

Теперь мы знаем, как создавать таблицы, так что создадим ещё одну с каталогом, однако все товары мы добавим уже через отдельную миграцию.

Здесь был очень долгий процесс описания каждого товара и поиск картинок для них. Карточка каждого товара генерировалась по шаблону, написанному ниже на скриншотах.

Фильтр работает на категориях, указанных у каждого товара, а также на отметке у отдельного столбца, относится ли товар к популярным.

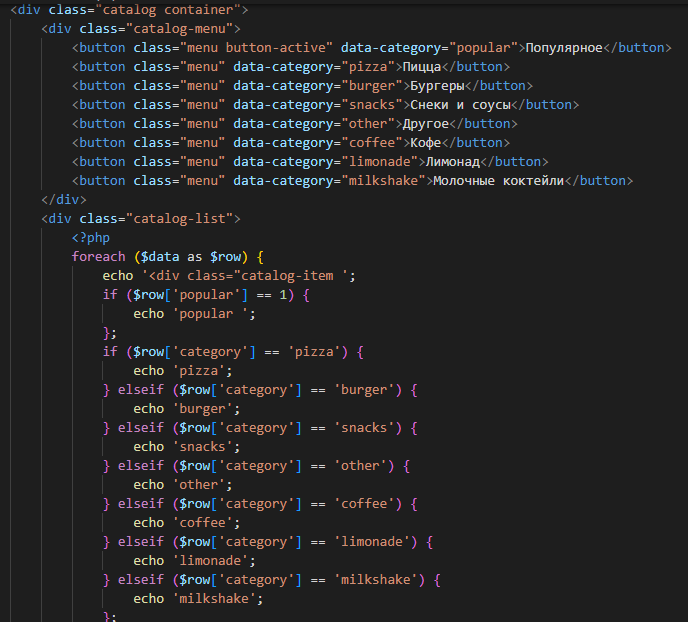


Рис. 3.3.1.

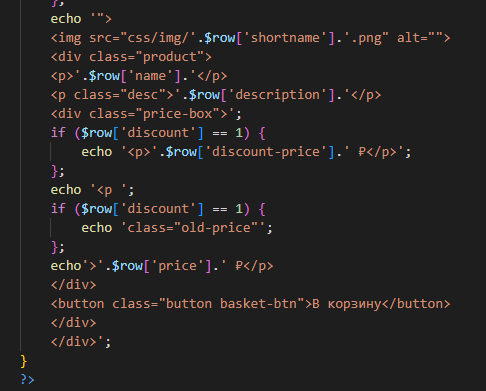


Рис. 3.3.2.