Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение

Высшего образования

“МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ”

(МОСКОВСКИЙ ПОЛИТЕХ)

Факультет информационных технологий

Кафедра Инфокогнитивные технологии

09.03.01 “Информатика и вычислительная техника”

Образовательная программа (профиль) “Веб-технологии”

По курсу Серверная веб-разработка, часть 1, курс 1, семестр 2

“Курсовой проект ”

Выполнил: Максимов Андрей Олегович

Группа:241-321

Проверил:

МОСКВА 2025

**Введение**

В современной веб-разработке использование фреймворков стало стандартом де-факто. Данный проект представляет особую ценность, так как:

- Позволяет глубоко понять архитектурные принципы MVC.

- Дает практический опыт реализации ключевых паттернов и шаблонов проектирования.

- Демонстрирует работу с реляционными базами данных через ORM.

**Цели, задачи**

1. Создание собственного фреймворка для создания веб приложений на базе языка программирования PHP.
2. Реализовать все паттерны проектирования указанные в курсе Серверной веб-разработки.
3. Создать базу данных для нашего веб приложения на базе фреймворка.
4. Реализовать возможность написания комментариев к статьям.

**Структура проекта**

Структура проекта была создана на основе паттерна проектирования MVC (сноска 1). А так же вспомогательных шаблонов ActiveRecord(сноска 2), Singlton(сноска 3), ORM(сноска 4).

СУБД для базы данных я взял MySQL.

Реализация связи с базой данных реализована при помощи сущности PDO.

**Этапы реализации**

1. Определение сущностей использующихся в паттерне проектирования MVC рис.6.
2. Создание файлов index.php, route.php, .htdaccess
3. Реализация в этих файлах системы перенаправления.
4. Получение route из строки поиска браузера.
5. Обработка этих route с помощью регулярного выражения.
6. Динамический вызов класса контролера и его метода в index.php.
7. Динамическое подключение классов с помощью spl\_autoload\_register
8. Прописывание все паттернов для роутинга в rout.php.
9. Активация текстового движка в файле конфигурации сервера APATCH.
10. Ввод команд для вывода в браузер нашего index.php при заходе в его корень.
11. Создание maincontroller.php и реализация в нем экшенов для main.php, hello.php и bye.php.
12. Создание базы данных.
13. Реализация конфига и класса DB.
14. Реализация в нем подключения к БД на базе объекта PDO.
15. Вынесение файла конфигурации подключения в отдельный файл settings.php.
16. Создание вида View.php.
17. Создание Articlecontroller для реализации экшенов связанных с выводом 1 статьи.
18. Создание модели Article для реализации вывода из таблички articles 1 строки.
19. Реализация экшена для вывода статьи.
20. Реализация экшенов и видов для реализации удаления, добавления и редактирования статей.
21. Создание модели ActiveRecordEntity.php для реализации паттерна ActiveRecord.
22. Создание в нем статических методов для выполнения запросов в бд.
23. Создание общих на все модели и обстрактных методов в ActiveRecordEntity.php.
24. Создание под-методов для метода save чтобы при сохранении в БД программа динамически определяла, нужно ли обновить запись в таблице или создать новую.
25. Создание модели User для хранения информации о пользователе.
26. Реализация паттерна Singltone в классе DB (private \_\_constract()).
27. Создание Commentcontroller для реализация возможности комментировать статьи.
28. Создание модели описывающей комментарий.
29. Доработка метода delete в Articlecontroller для удаления из БД всех комментариев к статье.
30. Создание статического метода в ActiveRecordEntity.php для реализации удаления всех комментариев к статье.
31. Реализация методов add, edit, delete в контроллере для комментариев с помощью уже созданных статических методов в ActiveRecordEntity.php.
32. Добавление css для стилизации веб-приложения.

**Описание функционала созданного веб-приложения**

Возможность просматривать созданные статьи рис.1.

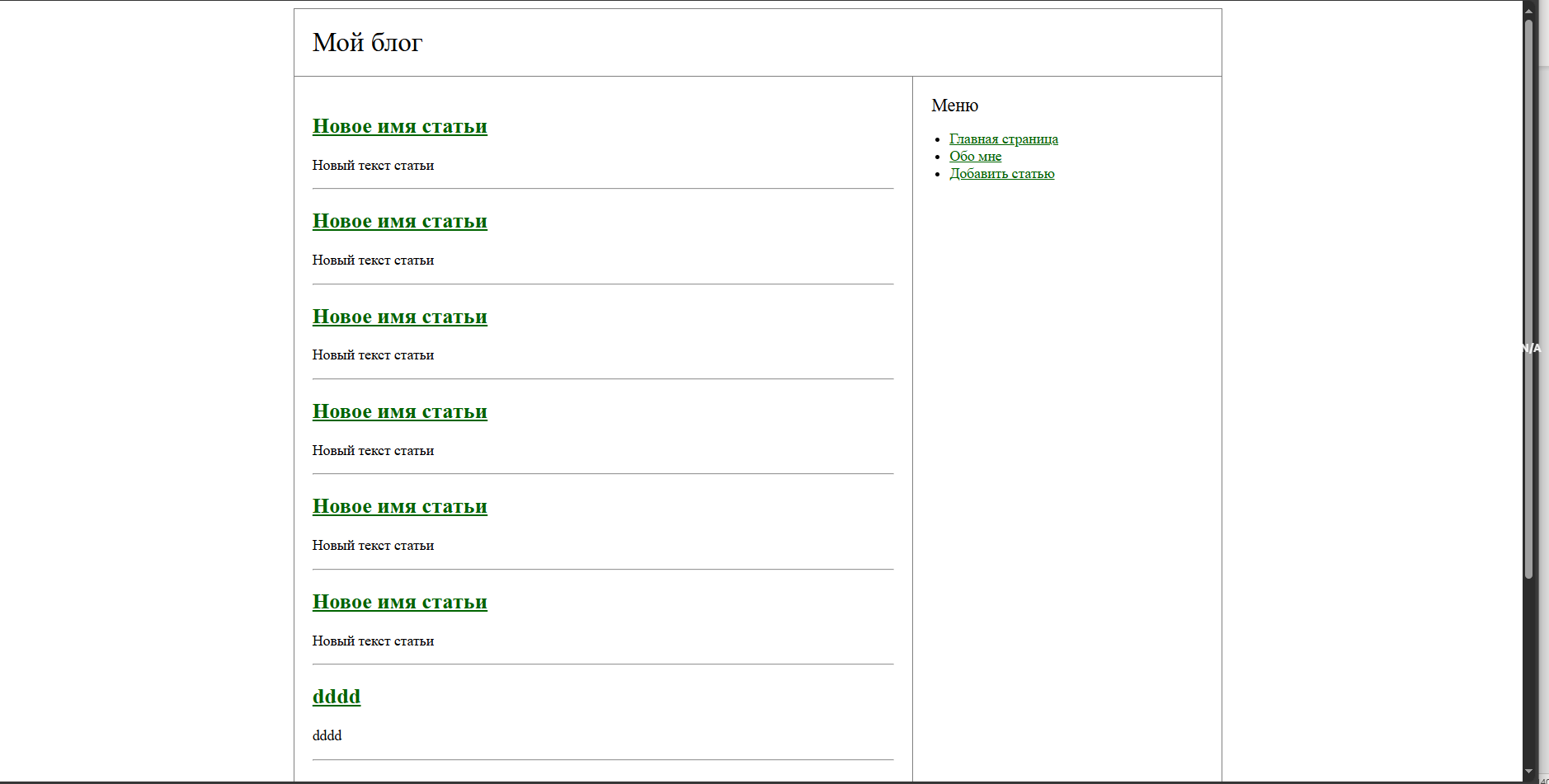


Рисунок 1 – Главная страница

Возможность заходить на страницу статьи при клике не нее рис.2.

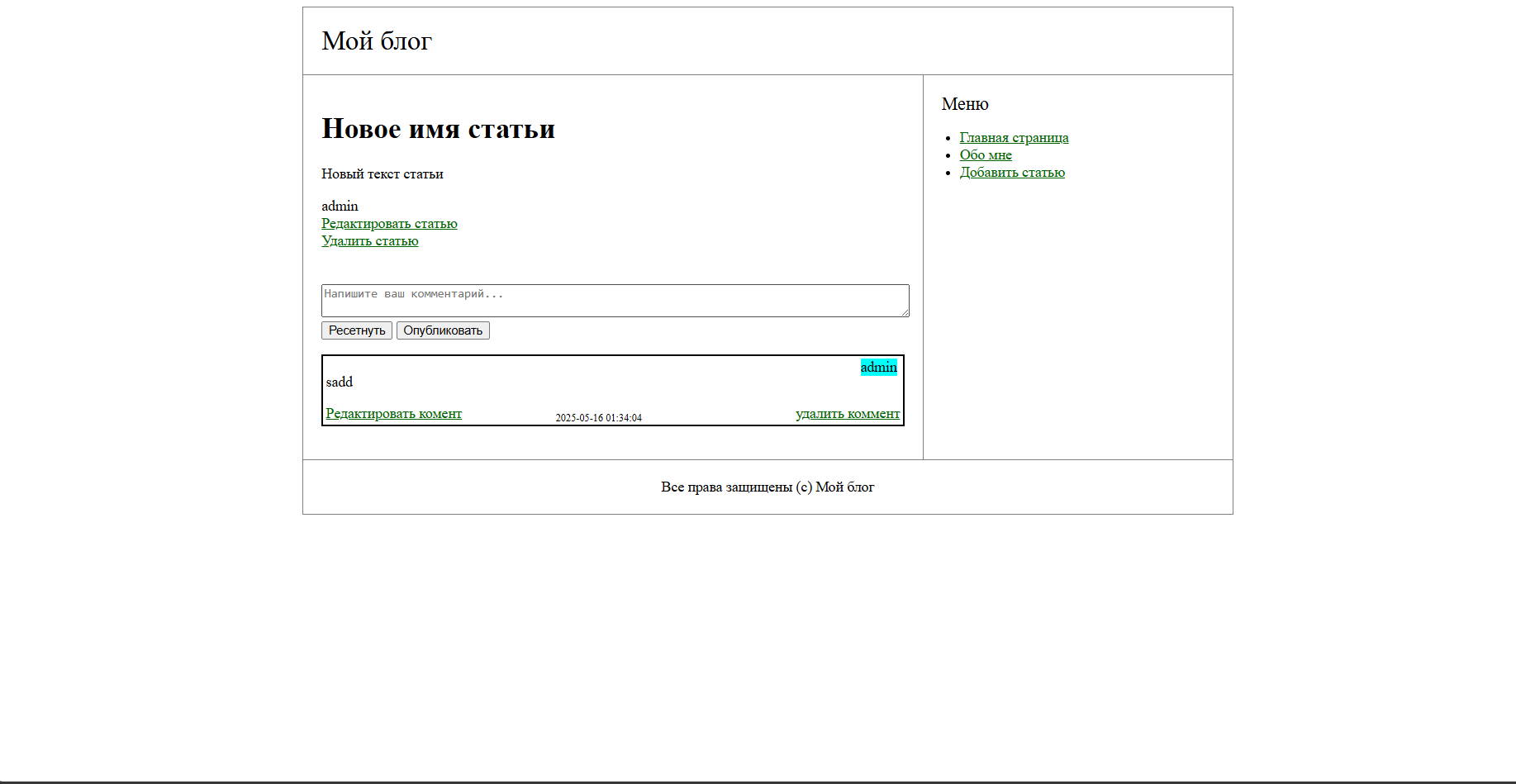


Рисунок 2 – страница статьи

Возможность добавлять статьи при клике на соответствующую кнопку рис.3.

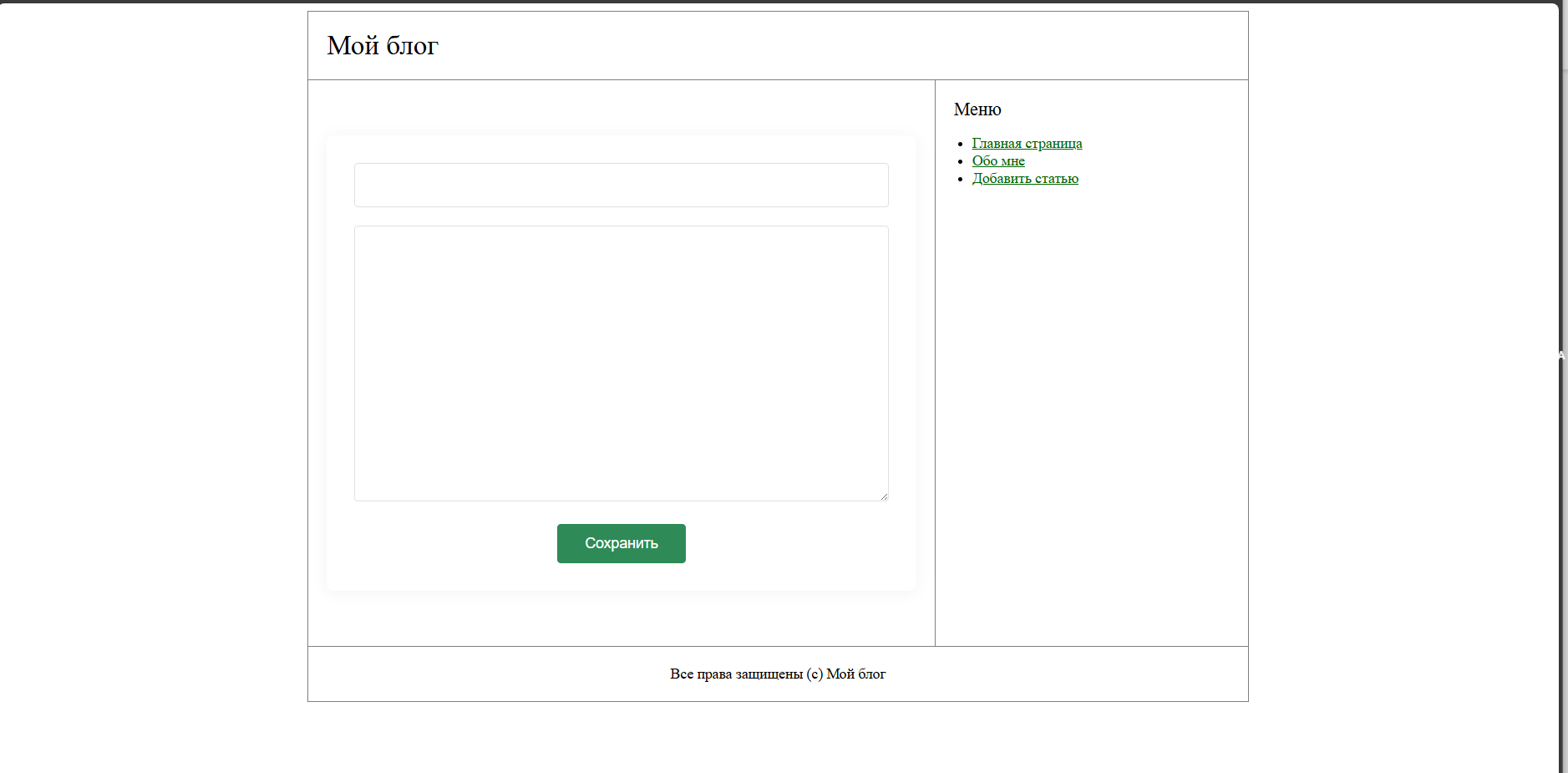


Рисунок 3 – страница создания статьи

Возможность удалять статью а вместе с ней и все комментарии относящиеся к ней.

Возможность оставлять комментарии рис.4.1 и 4.2.

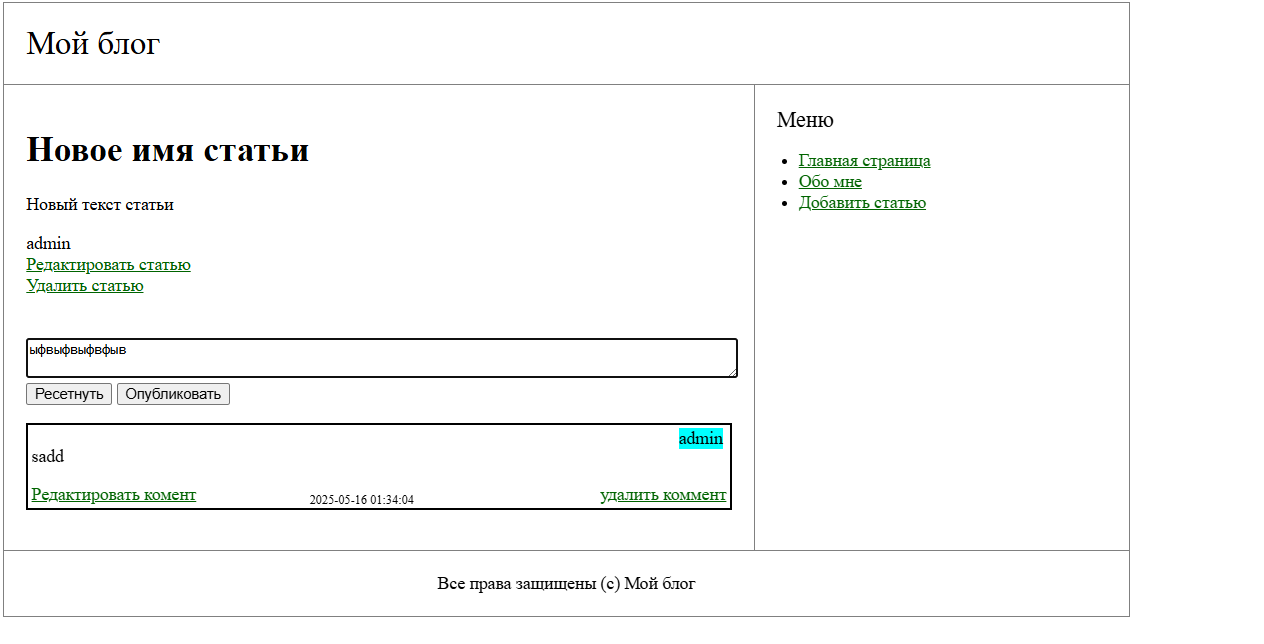


Рисунок 4.1 – создание комментария

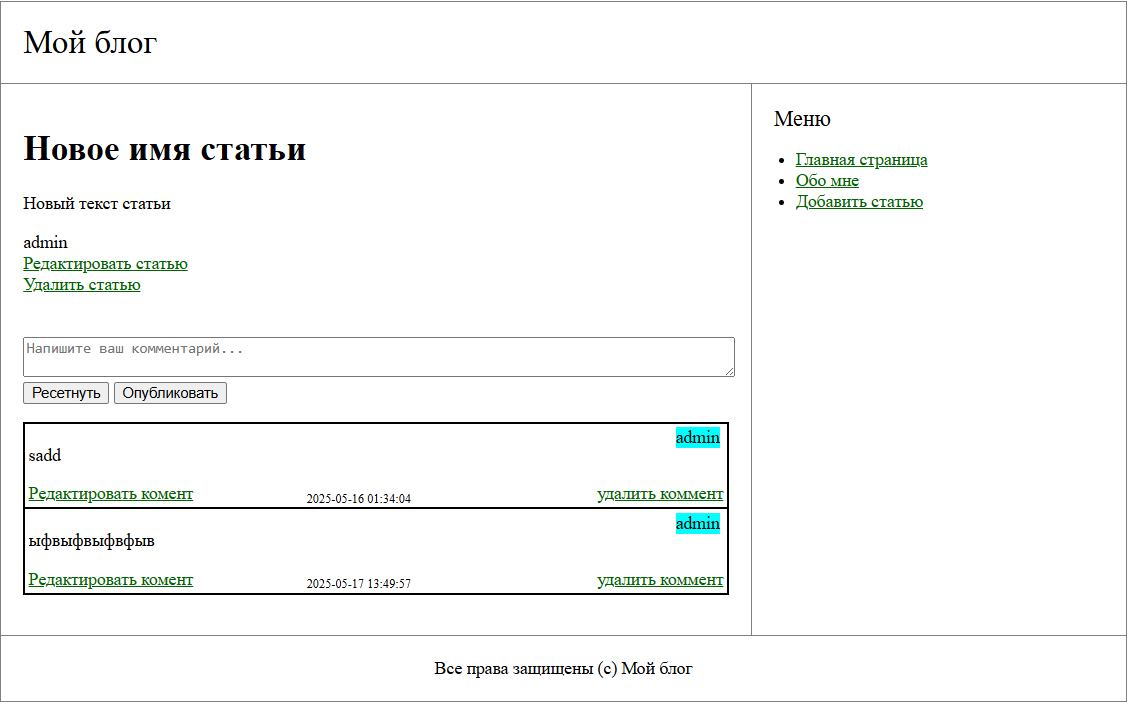


Рисунок 4.2 – создание комментария

Возможность изменять комментарии и удалять их рис.5.

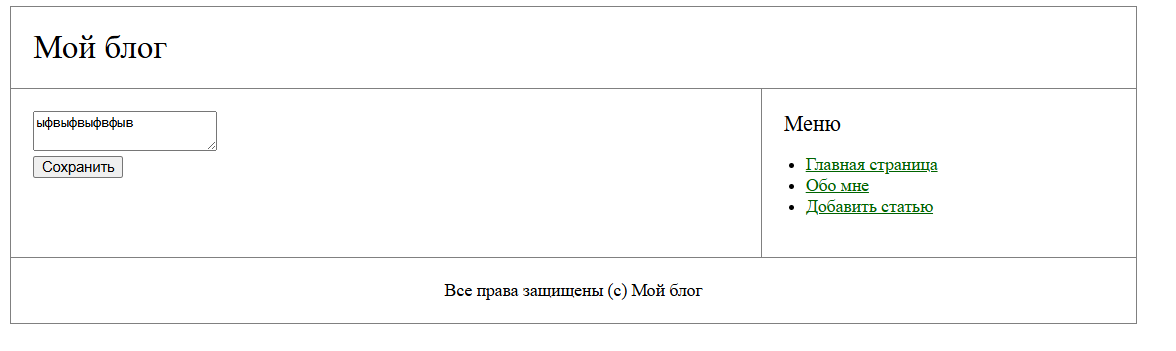


Рисунок 5 – Изменение комментария

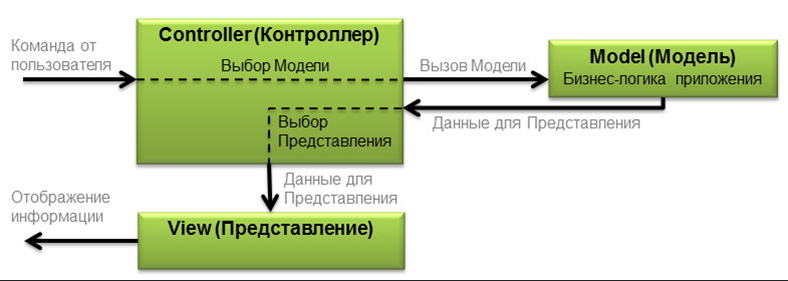


Рисунок 6 – диаграмма сущностей MVC.

**Вывод**

Вывод: в ходе курсового проекта по созданию фреймворка для создания веб-приложений на базе PHP я провел анализ паттерна MVC, познакомился с шаблонами проектирования ActiveRecordEntity, Singlton, ORM. Я создал свою DB и веб приложение на основе этого фреймворка. Смог более уверенно овладеть навыками программирования на PHP.

**Сноски**

сноска 1:

архитектурный паттерн, который разделяет приложение на три компонента. Цель — сделать код более понятным и разделить ответственность.

сноска 2:

шаблон проектирования приложений, описанный Мартином Фаулером в книге «Шаблоны архитектуры корпоративных приложений»

сноска 3:

шаблон проектирования, гарантирующий, что у класса будет только один экземпляр

сноска 4

шаблон проектирования, который позволяет связать данные с объектами в приложении