РЕФЕРАТ

Пояснительная записка 23 с., 8 рис., 6 табл., 7 источников, 3 прил.

САЙТ, СИСТЕМА УПРАВЛЕНИЯ КОНТЕТНОМ, КУРСОВОЙ ПРОЕКТ, ПРОГРАММНЫЙ ПРОДУКТ, PHP, HTML, CSS, JS, AJAX.

Целью курсового проекта является создание системы управления контентом RussianCraft Site Engine и запуск на ней сайта «Студенты.МГОК».

Проектирование и создание прототипа системы управления контентом (далее «CMS» или «движок») RCSE производилось в среде MS Visual Studio Code. Для создания макета страниц использовался Adobe Illustrator CC 2018. Пояснительная записка была реализована в MS Office Word 2016. Для разработки презентации была использована программа MS PowerPoint 2016. Блок-схемы были созданы в MS Office Visio 2016.

Предмет исследования – технология разработки универсальных и адаптивных веб-приложений.

Результатом работы стал программный продукт «Система управления контентом RussianCraft Site Engine».

Степень внедрения – будет использоваться в сети Интернет для создания сайтов.

Автор проекта подтверждает, что приведённый в курсовом проекте расчётно-аналитический материал правильно и объективно отражает состояние разрабатываемого объекта, все заимствованные из литературных и других источников теоретические и методологические положения и концепции сопровождаются ссылками на их авторов.

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 4](#_Toc531849303)

[1 Анализ предметной области и организации 6](#_Toc531849304)

[1.1 Характеристика и структура организации 6](#_Toc531849305)

[1.2 Анализ предметной области и существующих решений 7](#_Toc531849306)

[1.2.1 WordPress 9](#_Toc531849307)

[1.2.2 Joomla! 10](#_Toc531849308)

[1.2.3 Drupal 11](#_Toc531849309)

[2 Проектирование CMS и макета 13](#_Toc531849310)

[2.1 Техническое задание 13](#_Toc531849311)

[2.2 Технологии разработки 13](#_Toc531849312)

[2.3 Проектирование CMS 14](#_Toc531849313)

[2.4 Проектирование макета сайта 30](#_Toc531849314)

[2.5 Разработка прототипа CMS 30](#_Toc531849315)

[2.6 Разработка макета сайта 30](#_Toc531849316)

[ЗАКЛЮЧЕНИЕ 32](#_Toc531849317)

[СПИСОК ИСПОЛЬЗОВАНЫХ ИСТОЧНИКОВ 33](#_Toc531849318)

[ПРИЛОЖЕНИЕ А 34](#_Toc531849319)

[ПРИЛОЖЕНИЕ Б 35](#_Toc531849320)

[ПРИЛОЖЕНИЕ В 36](#_Toc531849321)

ВВЕДЕНИЕ

Несмотря на то, что современный Интернет появился относительно недавно – всего 27 лет назад – он уже глубоко интегрировался с нашей повседневной жизнью. Практически все новостные издания давно создали сетевые версии своих программ, основная часть общения приходится на сетевые сервисы, вроде WhatsApp, Facebook, или VK.

Любая энциклопедическая информация доступна в сети, также можно найти любые способы отдыха и развлечения – музыка, фильмы, игры, книги – все к нашим услугам. И не стоит забывать про все ещё кажущиеся фантастическими системы умных домов (даже тостеры могут выйти в Интернет!) и даже городов (у последнего ещё очень много противников, а возможные проблемы такой системы показаны во множестве фильмов и игр, пример – Watch Dogs).

Но что объединяет огромное число сервисов, доступных в сети Интернет? Это наличие сайта – страницы или набора связанных страниц, содержащих ту или иную информацию, использующуюся этим сервисом. Когда-то на заре Интернетов сайт представлял собой набор простых HTML-документов, жёстко связанных друг с другом, и изменение или добавление информации на них было сложным и трудоёмким процессом. С развитием Интернета увеличивалось и количество информации в нем, из-за чего стали появляться CMS – системы управления контентом, которые упростили жизнь владельцев сайтов до безобразия – например, для создания новых страниц теперь нужны всего пара действий и 15 минут, а не вечера в текстовом редакторе.

Для дальнейшего упрощения проведём параллель CMS с автомобилем – данные (база данных, файлы настроек) будут аккумулятором и баками для бензина, масла и тормозной жидкости, обработчики всех этих данных – насосы, трубки и блок цилиндров, а панель управления – руль, педали, КПП и тормоза. Но весь этот набор будет бесполезен без кузова, салона и приборной панели – ими будет шаблон-макет, который CMS заполняет данными.

Для успешного тестирования и, возможно, дальнейшего применения было решено спроектировать на базе RCSE студенческий портал колледжа МГОК. Таким образом, целью курсового можно обозначить: проектирование CMS RCSE и проектирование студенческого портала Студенты.МГОК. Для успешного достижения поставленной цели необходимо выполнить следующие задачи:

* спроектировать базу данных и файлы конфигурации;
* спроектировать функциональную часть CMS;
* спроектировать макет-шаблон портала Студенты.МГОК.

1 Анализ предметной области и организации

1.1 Характеристика и структура организации

Качественное образование лежит в основе любой отрасли, будь то юриспруденция, информационные технологи, машиностроение, дизайн или медицина. И не смотря на все увеличивающийся процент автоматизации большинства отраслей, человек остаётся незаменим, в особенности программисты, без которых современные производства находились бы на уровне 70-80х годов.

ГБПОУ «МГОК» предоставляет студентам возможности получения передовых специальностей в наиболее востребованных отраслях. Колледж следит за событиями в отраслях, которые преподаёт, способствует движению WorldSkills Russia, а его студенты не редко занимают высокие места на пьедесталах WSR.

Учреждение управляется советом «Союза машиностроителей России», представителем которого является директор «МГОК». В подчинении директора находятся заместители, которые управляют начальниками отделов и заведующими кафедр. Последние, в свою очередь, управляют непосредственно педагогами (а также сами являются ими). Педагоги же управляют студентами – преподают дисциплины и курируют группы.

Необходимо заметить, что любой из этих сотрудников может управляться вышестоящими (например, директор может напрямую отдать распоряжение конкретному студенту или педагогу), но не наоборот, иными словами, управление идёт строго сверху вниз.

Данная иерархия представлена в соответствии с рисунком 1.1.

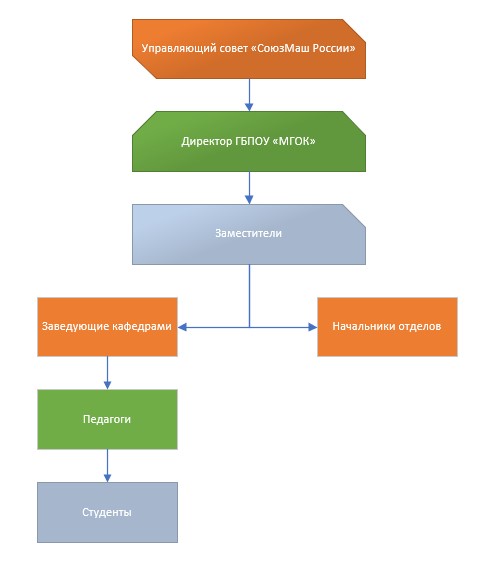
ц

Рисунок 1.1 – Структура организации

1.2 Анализ предметной области и существующих решений

Система управления контентом – информационная система или компьютерная программа, используемая для обеспечения и организации совместного процесса создания, редактирования и управления содержимым, иначе – контентом. CMS делятся на два подвида – Системы управления контентом масштаба предприятия (ECMS) и Системы управления веб-контентом (WCMS).

Из-за специфичности ECMS, под общим термином CMS как правило понимается WCMS. Рынок этих программных решений начал формироваться практически вместе с современным Интернетом в середине 90х и с тех пор рынок успел устояться и приобрести стабильных лидеров, всего известно свыше 1200 CMS [1].

Может возникнуть вопрос – а зачем, собственно, создавать ещё одну систему управления, если их так много? На разработку новой системы можно найти немало причин:

* создание нового продукта может привести к новым открытиям;
* новая система может быть компактней и быстрее, сохранив функционал более ранних решений;
* код нового решения может быть чище и проще, что позволит специалистам лучше понимать его, а также, эта причина может вызвать улучшение стабильности и снижение ошибок;
* если организация создаёт новое решение исключительно для внутреннего использования, то специалист, ответственный за продукт сможет точно отслеживать все проблемы продукта и вносить функционал, нужный этой организации.

На современном рынке CMS решений наиболее популярными по праву являются:

* «WordPress» – порядка 32,5% всех сайтов в сети;
* «Joomla!» – порядка 3% всех сайтов;
* «Drupal» – порядка 1,9% всех сайтов.

Также, около 45,5% сети либо не используют CMS, либо используют собственные решения [2].

Рассмотрим этих трёх лидеров и разберёмся, почему они лучшие и чему можно у них научиться.

1.2.1 WordPress

Выпущенный в мае 2003 года, WordPress является наиболее распространённой CMS в сети – более трети сайтов работают на нем. Причинами всенародной любви у этой системы можно назвать:

* система тем: благодаря ей системный администратор может легко сменить оформление своего сайта, не вмешиваясь в работу движка каким-либо образом;
* система плагинов: если сайту недостаточно функционала CMS, администратор может выбрать готовое расширение из крайне большого списка, или же написать свое;
* удачная архитектура: движок стабилен, быстр, не требователен к ресурсам, благодаря чему может работать без проблем даже на серверах десятилетней давности;
* open-source: движок распространяется по лицензии GPL 2, что позволяет разработчикам создавать свои версии этой CMS и выпускать их как собственное решение.

Как можно видеть, WordPress не использует каких-либо ноу-хау: все элементы архитектуры этой CMS присутствуют и практически во всех альтернативах – Панель управления, Обработчики баз данных, файлов настроек, Система логгирования, Парсер шаблонов, Plugin API.

Также, WordPress располагает интуитивно понятным, простым интерфейсом: все настройки и функции распределены по категориям, все названия сопровождаются говорящими иконками. В дополнение к этому, для управления сайтом не требуются знания английского: все названия переведены на русский, что серьезно облегчает жизнь. Интерфейс панели управления WordPress представлен на рисунке 1.2.

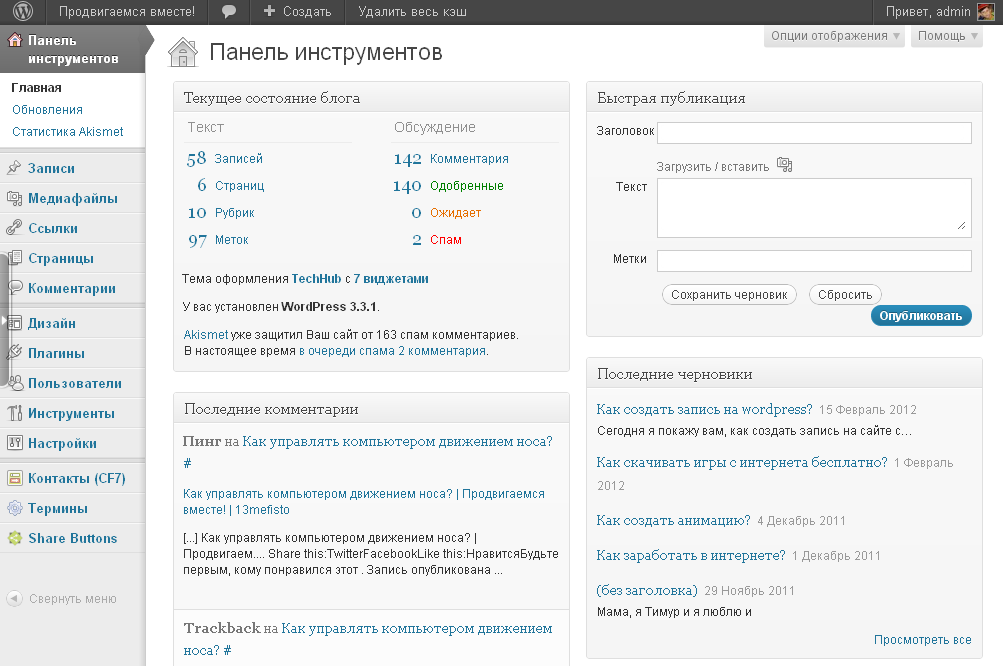


Рисунок 1.2 – Панель управления WordPress

1.2.2 Joomla!

Вышла в сентябре 2005 как ответвление CMS Mambo с исправлениями известных ошибок родительской CMS. Движок прочно обосновался на 2м месте по популярности, но отрыв от первого места настолько велик, что шансы на выход на первое место минимальны. Основные «фишки» движка такие же, как и у WordPress. Основными отличиями от WordPress являются:

* возможность редактировать front- и back-end прямо в панели управления;
* модульный шаблон: можно настроить структуру страницы и сохранить ее.

Система Joomla, не смотря на функциональное преимущество, имеет и перечень недостатков:

* меньшее, по сравнению с WordPress сообщество, что приводит к меньшему количеству помощи в случае проблем;
* стоимость разработки проекта на Joomla будет выше, чем на WordPress;
* интуитивность интерфейса панели управления Joomla меньше, чем у WordPress (Панель управления представлена на рисунке 1.3).

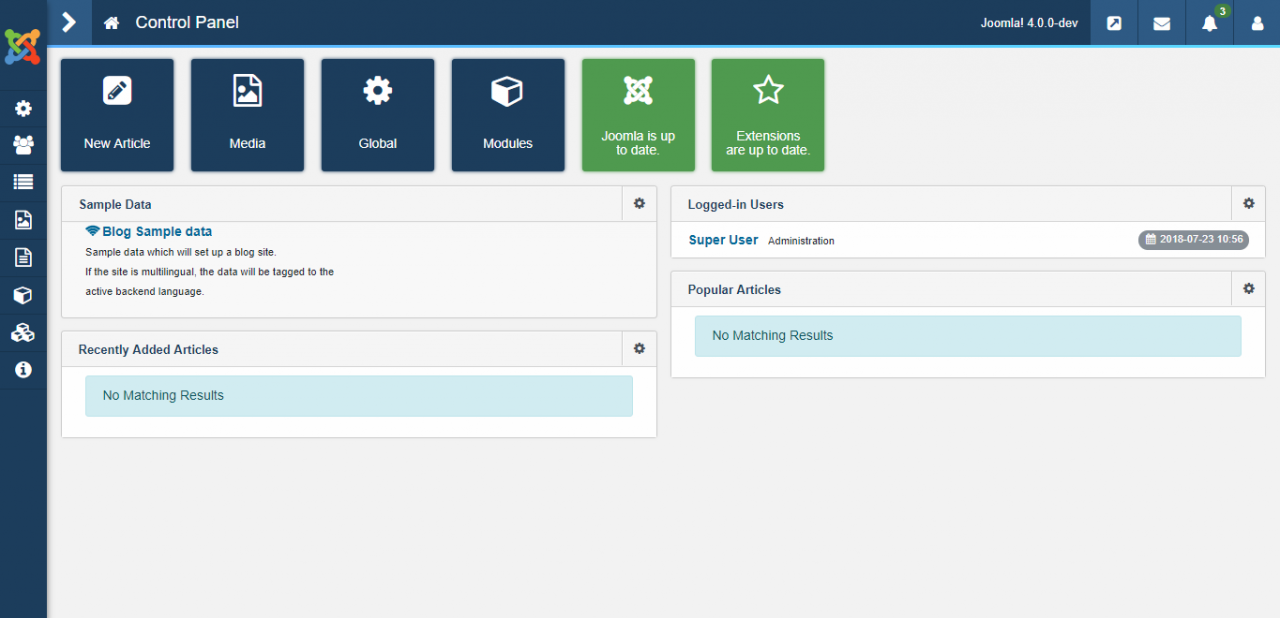


Рисунок 1.3 – Панель управления Joomla

1.2.3 Drupal

Впервые в сети движок засветился в 2000 году, как безымянная основа для управления небольшой локальной сетью. Помимо сайтов, Drupal позволяет создавать веб-приложения. Несмотря на постоянную команду разработки (10 человек), немалый вклад в развитие CMS вносится сообществом.

Аналогично Joomla! движок также имеет все основные «фишки» WordPress, и также имеет отличия:

* поддержка XML;
* механизм авторизации на основе OpenID;
* поддержка мультисайтинга – нескольких наборов настроек на одном хосте.

Специалисты выделяют два основных недостатка Drupal:

* слабое использование объектно-ориентированных принципов;
* отсутствие обратной совместимости API со старыми версиями.

Также Drupal имеет не самую удобную Панель управления (представлена на рисунке 1.4).

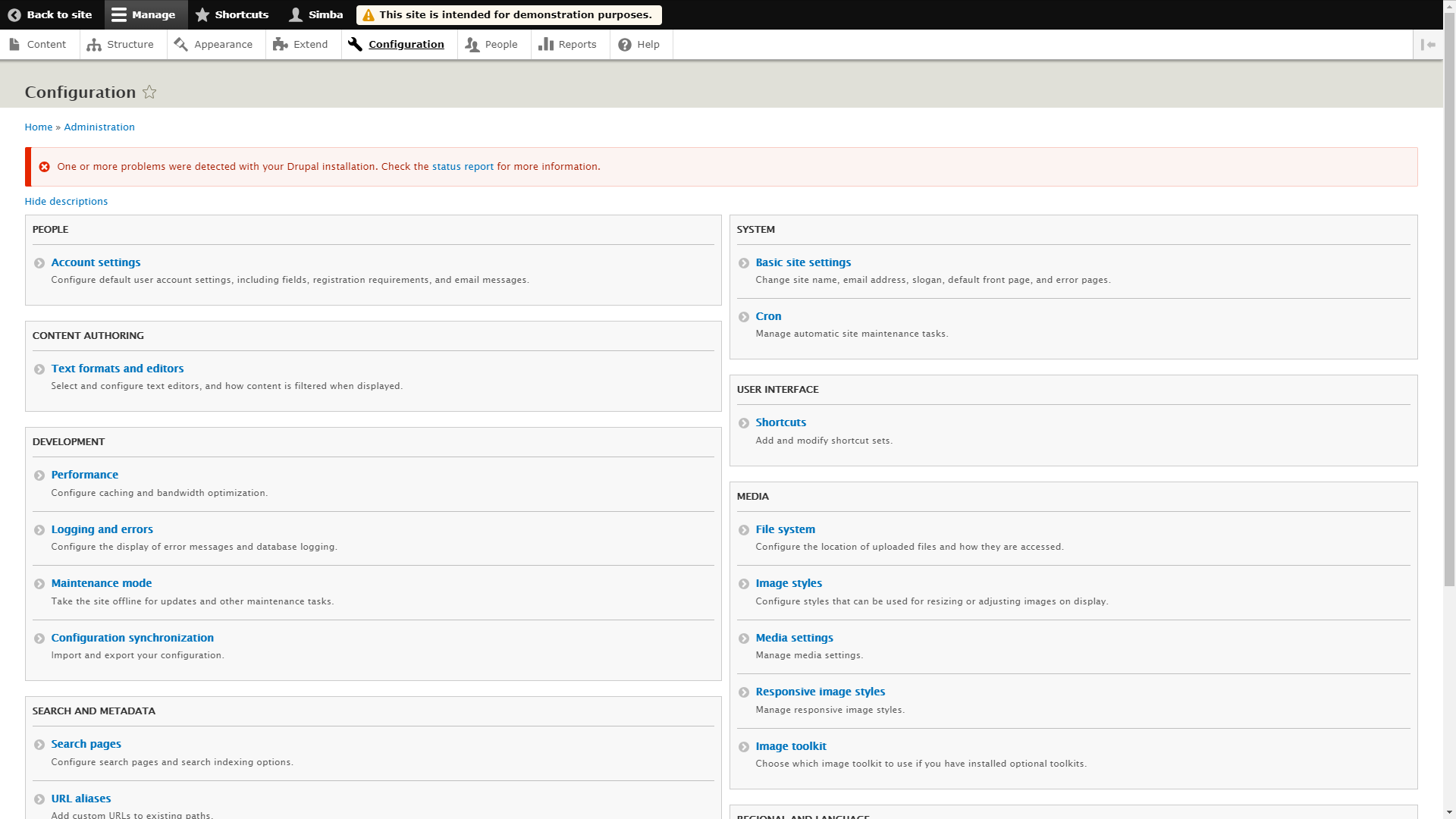


Рисунок 1.4 – Панель управления Drupal

2 Проектирование CMS и макета

2.1 Техническое задание

В данной работе необходимо разработать систему управления контентом «RussianCraft Site Engine» и запустить на ней студенческий портал для ГБПОУ МГОК. Для этого сначала необходимо определить требования, которым должен удовлетворять создаваемый ресурс, затем необходимо определить технологии и средства, с помощью которых будет создаваться ресурс. После этого можно приступить к разработке. Сначала необходимо функциональную архитектуру и разработать макет сайта.

После завершения разработки необходимо проверить функциональность сайта, наполнить его контентом и разместить его в сети Интернет.

Подробное техническое задание представлено в Приложении А.

2.2 Технологии разработки

Для разработки использованы следующие языки программирования и разметки:

1. PHP - скриптовый языкобщего назначения, интенсивно применяемый для разработки веб-приложений. В настоящее время поддерживается подавляющим большинством хостинг-провайдеров и является одним из лидеров среди языков, применяющихся для создания динамических веб-сайтов [3].
2. HTML – стандартизированный язык разметки документов во Всемирной паутине. Большинство веб-страниц содержат описание разметки на языке HTML (или XHTML). Язык HTML интерпретируется браузерами; полученный в результате интерпретации форматированный текст отображается на экране монитора компьютера или мобильного устройства [4].
3. CSS - формальный язык описания внешнего вида документа, написанного с использованием языка разметки. Преимущественно используется как средство описания, оформления внешнего вида веб-страниц, написанных с помощью языков разметки HTML и XHTML, но может также применяться к любым XML-документам [5].

Для написания кода программы применён MS Visual Studio Code.

MS Visual Studio Code — редактор исходного кода, разработанный Microsoft для Windows, Linux и macOS. Позиционируется как «лёгкий» редактор кода для кроссплатформенной разработки веб- и облачных приложений. Включает в себя отладчик, инструменты для работы с Git, подсветку синтаксиса, IntelliSense и средства для рефакторинга. Имеет широкие возможности для кастомизации: пользовательские темы, сочетания клавиш и файлы конфигурации. Распространяется бесплатно, разрабатывается как программное обеспечение с открытым исходным кодом, но готовые сборки распространяются под проприетарной лицензией [6].

Для проектирования базы данных использована программа MySQL Workbench.

MySQL Workbench – инструмент для визуального проектирования баз данных, интегрирующий проектирование, моделирование, создание и эксплуатацию БД в единое бесшовное окружение для системы баз данных MySQL [7].

2.3 Проектирование CMS

Для начала необходимо построить общую схему движка, чтобы получить представление о модулях, которые необходимо разработать для получения требуемого функционала. Также будет необходимо построить схемы каждого модуля, чтобы упростить и ускорить разработку.

Общая схема разделяется на уровни относительно хранилища данных: на первом уровне (Уровень 0) находятся База данных, файлы конфигурации и логи. На втором уровне (Уровень 1) находится ядро движка – обработчики Базы данных, файлов конфигурации и логов. На третьем (Уровень 2) – обработчики данных для Системы пользователей, Форума, Новостной ленты, Панели управления и Поиска. Четвёртый (Уровень 3) включает в себя конечные обработчики – Шаблоны и Plugin API. Пятый уровень (Уровень 4) включает в себя расширения для CMS – плагины. Шестой (Уровень 5) это конечная точка – браузер пользователя.

Примечание: из-за сложности реализации, Поиск и Plugin API на данном этапе решено не разрабатывать.

Схема представлена на рисунке 2.1.

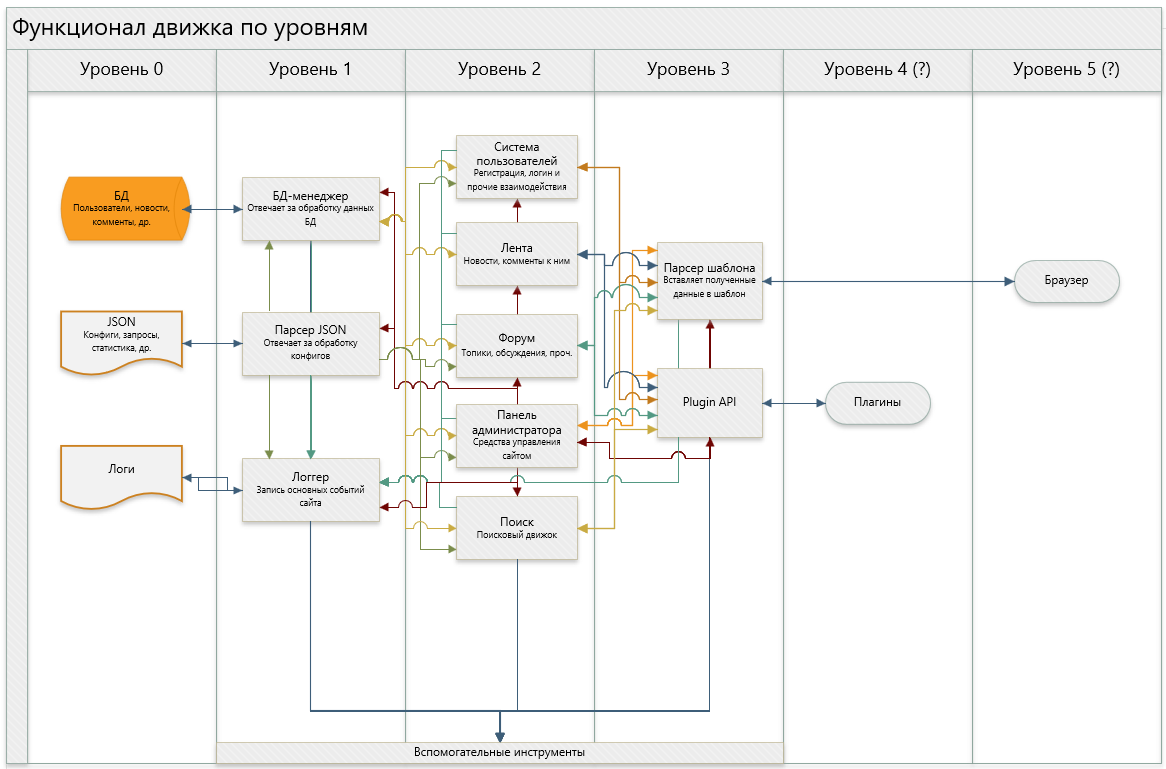


Рисунок 2.1 – Общая схема движка

Далее – структура базы данных. Для планируемого функционала потребуется 6 таблиц: таблица пользователей, таблица наказанных пользователей, таблица новостей, таблица комментариев к новостям, таблица тем на форумах и таблица ответов для этих тем.

В таблице пользователей – accounts – всего 8 полей: логин, пароль, почта, пол, дата рождения, город, дата регистрации и настройки. Все поля, кроме пола, даты рождения и города обязательны для заполнения.

Структура таблицы accounts представлена в таблице 2.1.

Таблица 2.1 – Структура таблицы accounts

|  |  |  |  |
| --- | --- | --- | --- |
| Ключ | Название | Тип данных | Длина |
| Первичный | login | varchar | 20 |
| - | password | varchar | 255 |
| - | email | varchar | 127 |
| - | sex | varchar | 1 |
| - | brithdate | varchar | 10 |
| - | origin | varchar | 255 |
| - | regdate | varchar | 10 |
| - | settings | text | - |

Таблица наказанных пользователей – punishments – содержит 6 полей: логин, тип наказания, причина, правило, дата, дата истечения. Все поля в этой таблице обязательны для заполнения.

Структура этой таблицы представлена в таблице 2.2.

Таблица 2.2 – Структура таблицы punishments

|  |  |  |  |
| --- | --- | --- | --- |
| Ключ | Название | Тип данных | Длина |
| Первичный, внешний | login | varchar | 20 |
| - | type | varchar | 8 |
| - | reason | text | - |
| - | rule | varchar | 8 |

Продолжение таблицы 2.2

|  |  |  |  |
| --- | --- | --- | --- |
| - | date | varchar | 15 |
| - | expires | varchar | 15 |

Таблица новостных записей – posts – содержит 10 полей: идентификатор записи, заголовок, дату, автора, голоса за, голоса против, описание, теги, текст новости, настройки. Все поля кроме описания и тегов обязательны для заполнения.

Структура таблицы представлена в таблице 2.3.

Таблица 2.3 – Структура таблицы posts

|  |  |  |  |
| --- | --- | --- | --- |
| Ключ | Название | Тип данных | Длина |
| Первичный | post\_id | int | - |
| - | title | varchar | 45 |
| - | date | varchar | 16 |
| Внешний | author | varchar | 20 |
| - | voteups | int | - |
| - | votedowns | int | - |
| - | description | varchar | 127 |
| - | tags | text | - |
| - | content | text | - |
| - | settings | text | - |

Таблица комментариев к новостям – comments – содержит 8

полей: идентификатор комментария, идентификатор родителя, дату, автора, голоса за, голоса против, текст комментария, настройки. Все поля обязательны для заполнения.

Структура таблицы представлена в таблице 2.4.

Таблица 2.4 – Структура таблицы comments

|  |  |  |  |
| --- | --- | --- | --- |
| Ключ | Название | Тип данных | Длина |
| Первичный | comment\_id | int | - |
| Внешний | reply\_to | int | - |
| - | date | varchar | 19 |
| Внешний | author | varchar | 20 |
| - | voteups | int | - |
| - | votedowns | int | - |
| - | content | text | - |
| - | settings | text | - |

Таблица тем на форуме – topics – содержит 10 полей: идентификатор записи, заголовок, раздел, дату, автора, голоса за, голоса против, теги, текст записи, настройки. Все поля таблицы кроме тегов обязательны для заполнения.

Структура таблицы представлена в таблице 2.5.

Таблица 2.5 – Структура таблицы topics

|  |  |  |  |
| --- | --- | --- | --- |
| Ключ | Название | Тип данных | Длина |
| Первичный | topic\_id | int | - |
| - | title | varchar | 45 |
| - | section | varchar | 45 |
| - | date | varchar | 16 |
| Внешний | author | varchar | 20 |
| - | voteups | int | - |
| - | votedowns | int | - |
| - | tags | text | - |
| - | content | text | - |
| - | settings | text | - |

Таблица ответов на темы – replies – содержит 8 полей: идентификатор комментария, идентификатор родителя, дату, автора, голоса за, голоса против, текст комментария, настройки. Все поля обязательны для заполнения.

Структура таблицы представлена в таблице 2.6.

Таблица 2.6 – Структура таблицы comments

|  |  |  |  |
| --- | --- | --- | --- |
| Ключ | Название | Тип данных | Длина |
| Первичный | reply\_id | int | - |
| Внешний | reply\_to | int | - |
| - | date | varchar | 19 |
| Внешний | author | varchar | 20 |
| - | voteups | int | - |
| - | votedowns | int | - |
| - | content | text | - |
| - | settings | text | - |

Также, все таблицы необходимо объединить в ER-диаграмму, указав связи между полями-ключами в таблицах.

Диаграмма представлена на рисунке 2.2.

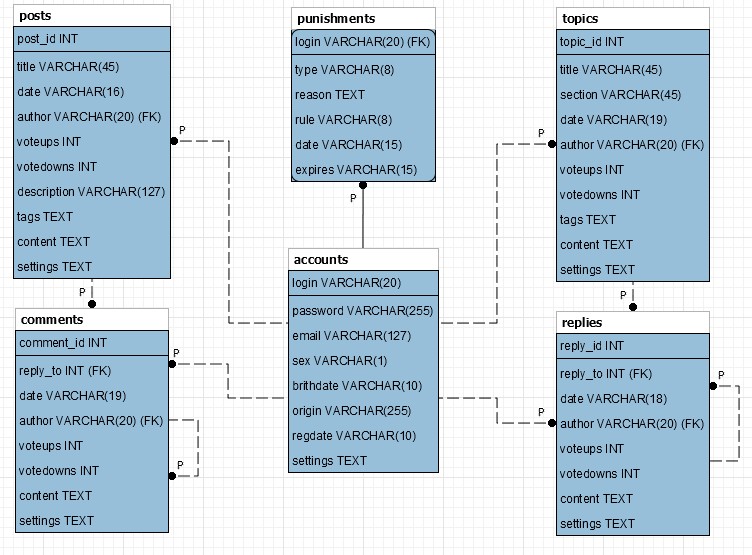


Рисунок 2.2 – ER-диаграмма базы данных

Далее необходимо построить структуру основных файлов конфигурации. Данные файлы имеют формат JSON, что облегчает структуризацию данных. Для основного функционала необходимы следующие файлы:

* main.json: содержит основные настройки сайта и параметры подключения к базе данных. Структура файла представлена на рисунке 2.3.
* queries.json: содержит шаблоны запросов к базе данных для каждой таблицы. Структура файла представлена на рисунке 2.4.
* usergroups.json: содержит параметры групп пользователей, в которых указаны права и возможности по основным категориям функционала. Структура файла представлена на рисунке 2.5.
* modules.json: содержит параметры для каждого модуля движка. Структура файла представлена на рисунке 2.6.
* ban\_types.json: содержит описания видов наказаний для нарушителей правил. Структура файла представлена на рисунке 2.7.
* forum\_sections.json: содержит параметры разделов форума. Структура файла представлена на рисунке 2.8.



Рисунок 2.3 – Структура файла main.json



Рисунок 2.4 – Структура файла queries.json

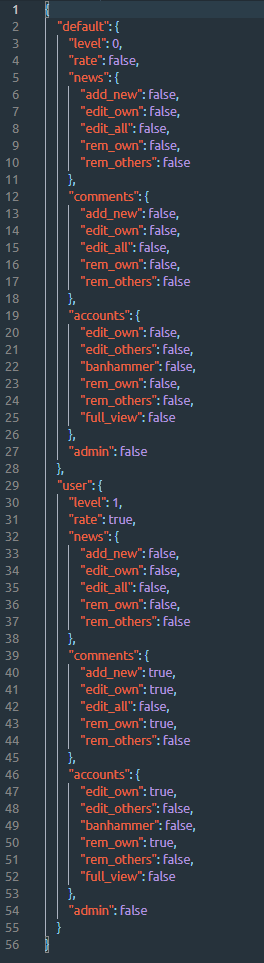


Рисунок 2.5 – Структура файла usergroups.json

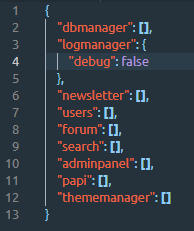


Рисунок 2.6 – Структура файла modules.json

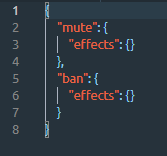


Рисунок 2.7 – Структура файла ban\_types.json

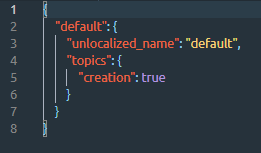


Рисунок 2.8 – Структура файла forum\_sections.json

Далее – схемы модулей движка. Каждый модуль представляет собой класс с набором методов, используемых им.

В основе всего лежит Обработчик конфигурации – JSONManager, отвечающий за обработку данных в формате JSON и передачу этих данных другим модулям. Структура класса JSONManager представлена на рисунке 2.9.

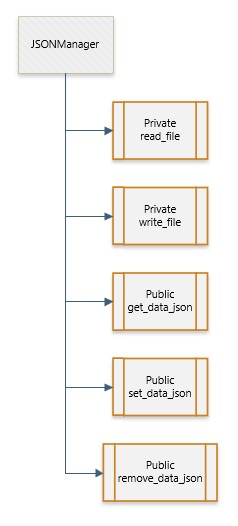


Рисунок 2.9 – Структура класса JSONManager

Следующий основополагающий элемент – Обработчик баз данных, DBManager. Он отвечает за загрузку\выгрузку данных из БД и передачу их другим модулям. Структура этого класса представлена на рисунке 2.10.

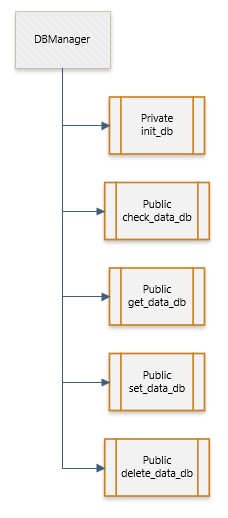


Рисунок 2.10 – Структура класса DBManager

Следующий элемент основной тройки модулей движка – Логгер или LogManager. Он ведёт запись всех важных событий внутри движка в файлы, чтобы в случае ошибки можно было без проблем узнать, где и что пошло не так. Структура класса представлена на рисунке 2.11.

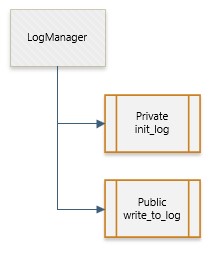


Рисунок 2.11 – Структура класса LogManager

Далее – главный узел взаимодействия с пользователями, класс UserManager. Он отвечает за обработку данных пользователя – регистрацию, вход, настройку аккаунта и так далее. Структура этого класса представлена на рисунке 2.12.

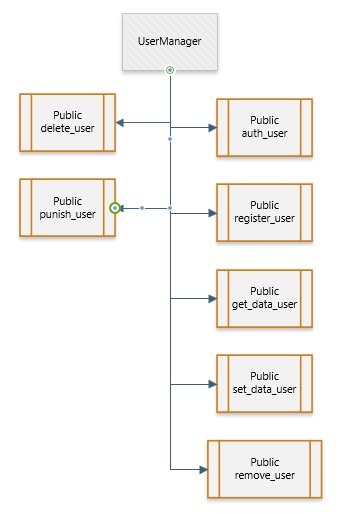


Рисунок 2.12 – Структура класса UserManager

Следующим идёт Обработчик новостной ленты, PostsManager. Он отвечает за новостную ленту – добавление, редактирование, загрузка и удаление новостных записей, а также за комментарии к ним. Структура класса представлена на рисунке 2.13.

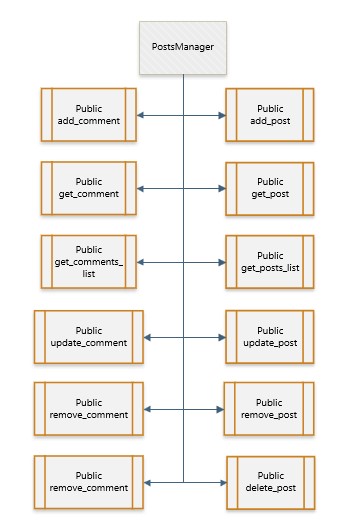


Рисунок 2.13 – Структура класса PostsManager.

Класс ForumManager вполне предсказуемо отвечает за работу форума – добавление новых обсуждений, редактирование и удаление существующих. Структура во многом похожа на PostsManager, и представлена на рисунке 2.14.

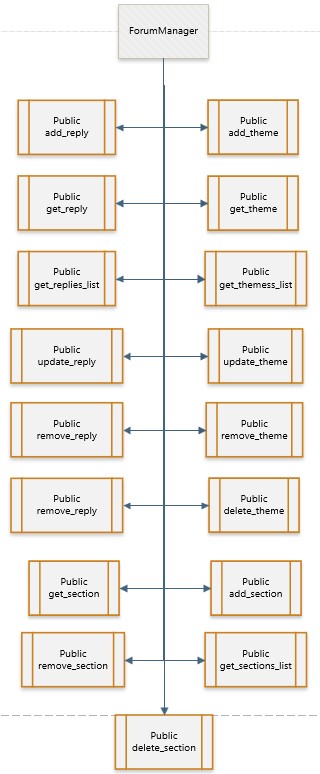


Рисунок 2.14 – Структура класса ForumManager

Панель управления или AdminPanel – немаловажный модуль CMS, является достаточно комплексной системой, поэтому её структура, представленная на рисунке 2.15, была упрощена.

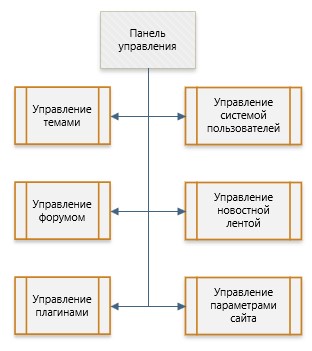


Рисунок 2.15 – Упрощённая структура AdminPanel

И последнее, но не по значимости – TemplateManager. Этот класс отвечает за вывод полученной предыдущими обработчиками информации в браузер пользователя. Структура его представлена на рисунке 2.16.

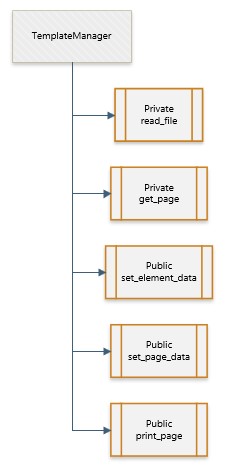


Рисунок 2.15 – Структура класса TemplateManager

2.4 Проектирование макета сайта

В начале необходимо создать общий макет страницы, который будет содержать базовые элементы, такие, как: шапка, меню, зона контента и подвал.

Общий макет представлен на рисунке 2.16.



Рисунок 2.16 – Общий макет страницы

После создания общего макета необходимо сделать базовый шаблон, который будет видоизменяться для адаптации под контент на странице. Макет такого шаблона представлен на рисунке 2.17.

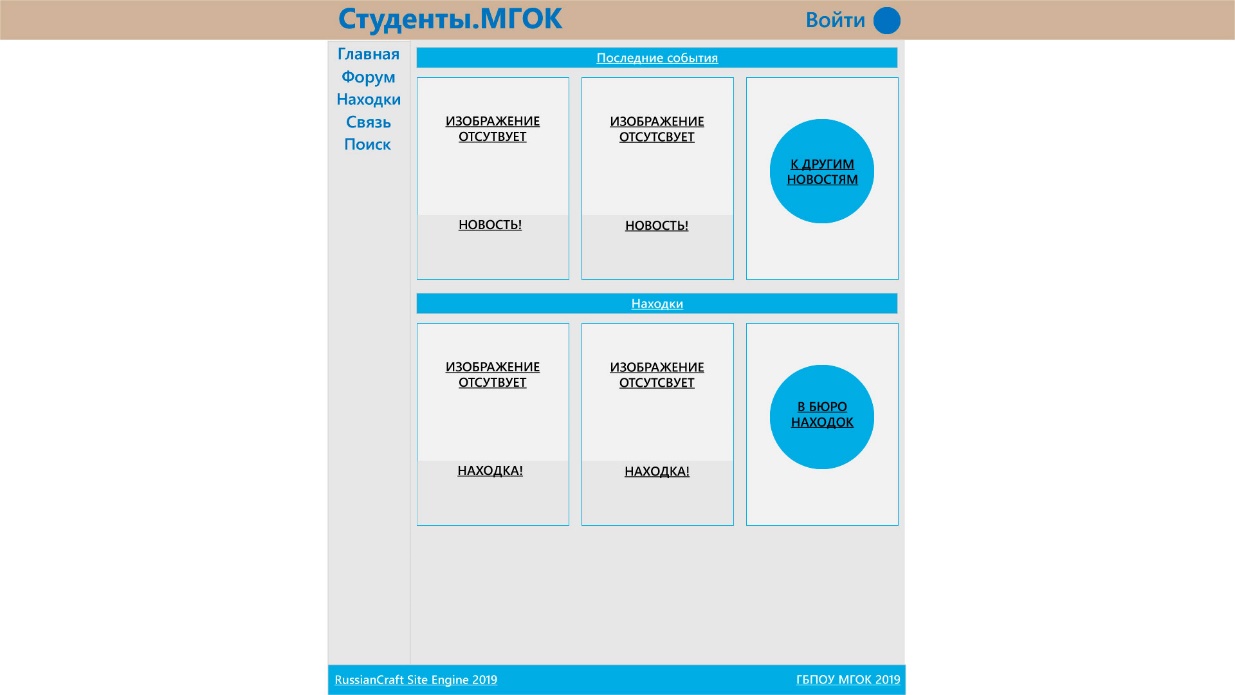


Рисунок 2.17 – Базовый шаблон сайта

2.5 Разработка прототипа CMS

Для начала необходимо создать базу данных основываясь на проекте, составленном ранее. Для проектирования и загрузки БД на сервер используется MySQL Workbench. Экран создания таблиц показан на рисунке 2.18.

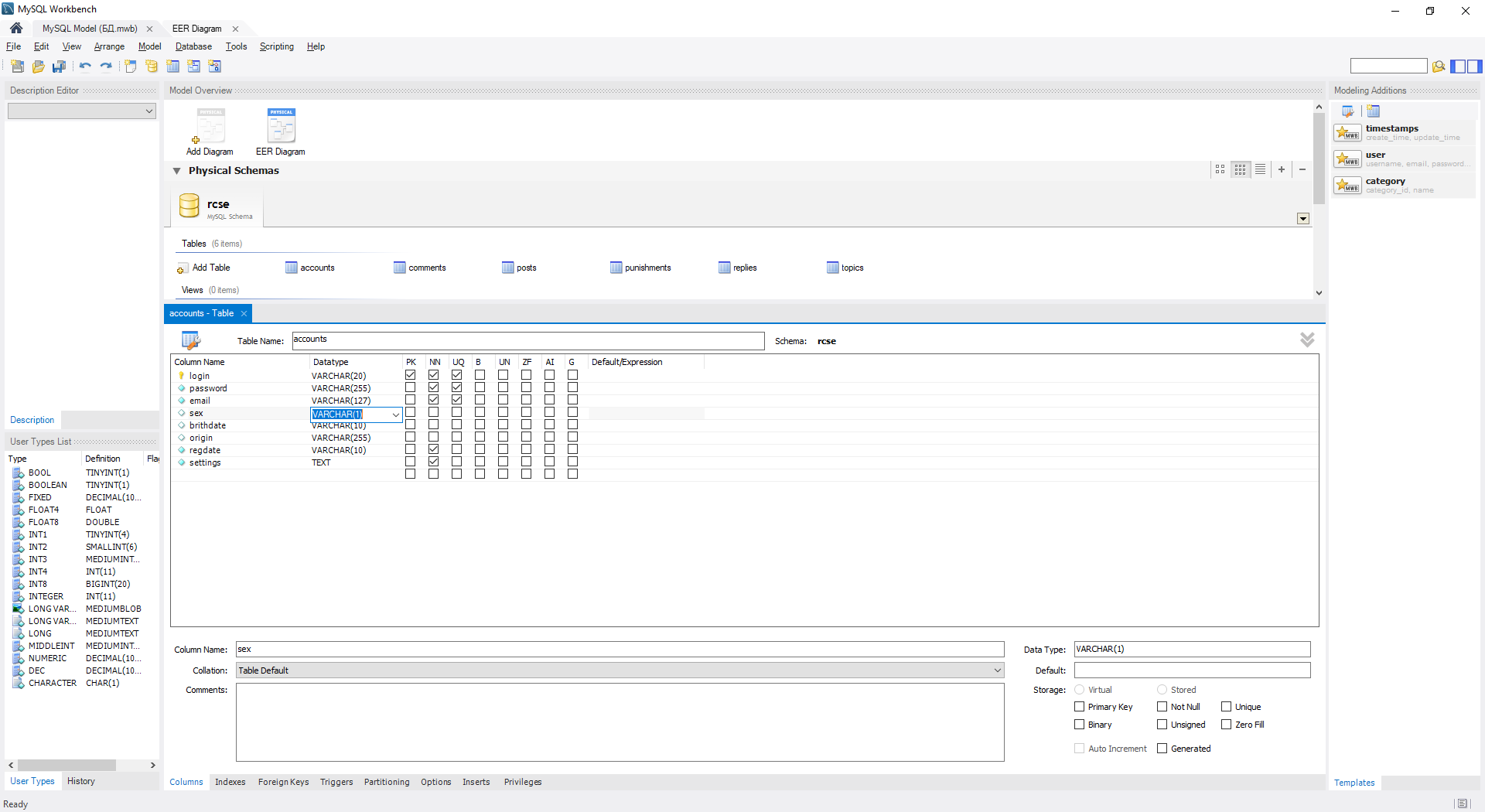


Рисунок 2.18 – Создание таблицы в MySQL Workbench

После создания всех нужных таблиц, их необходимо выстроить в единую ER-диаграмму. Экран создания ER-диаграмм показан на рисунке 2.19.

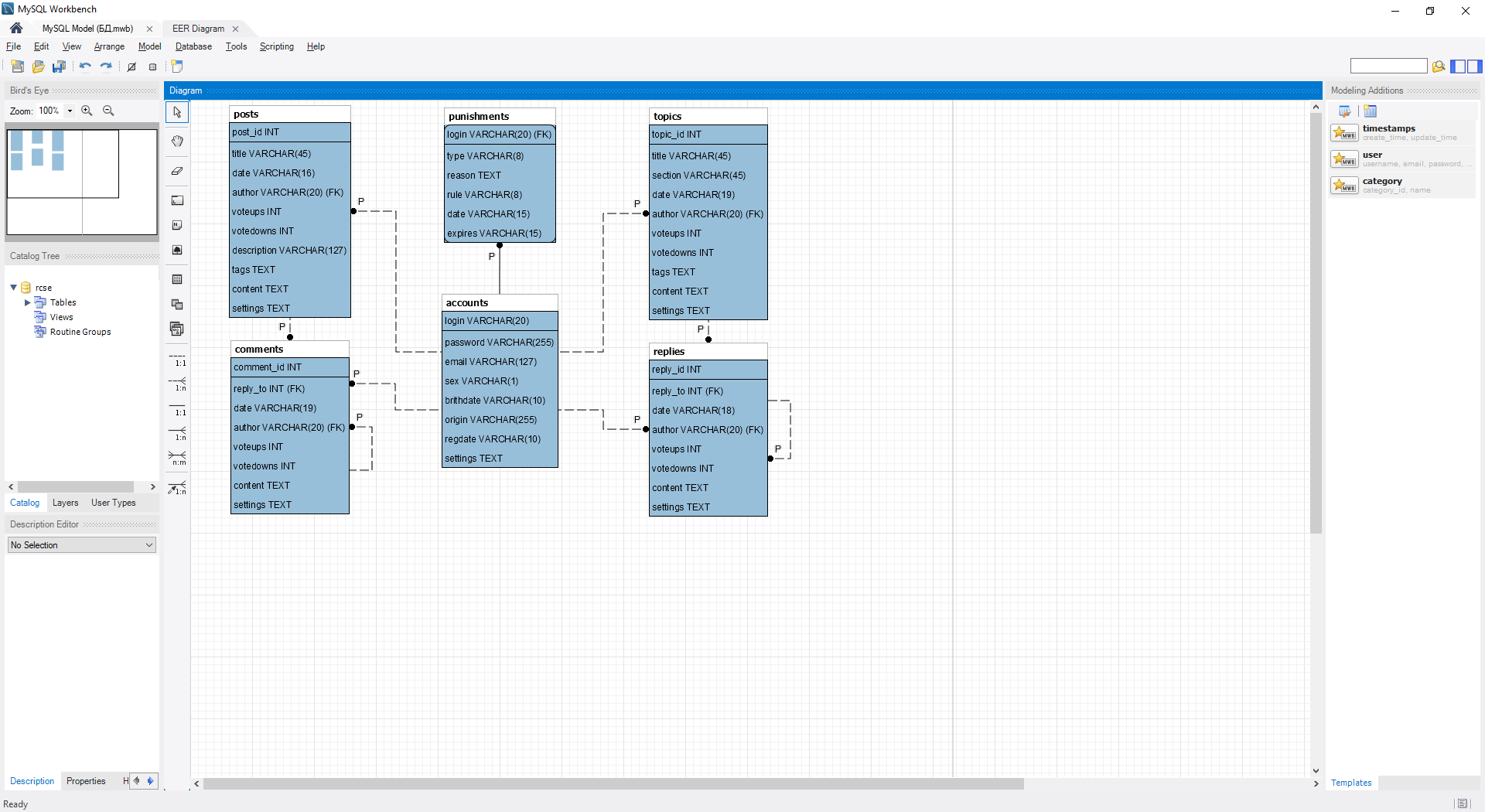


Рисунок 2.19 – Экран создания ER-диаграмм

После завершения создания базы данных, её необходимо загрузить в MySQL сервер для дальнейшего пользования. Делается это при помощи функции «Forward engineer» во вкладке «Database». Окно загрузки БД на сервер изображён на рисунке 2.20.

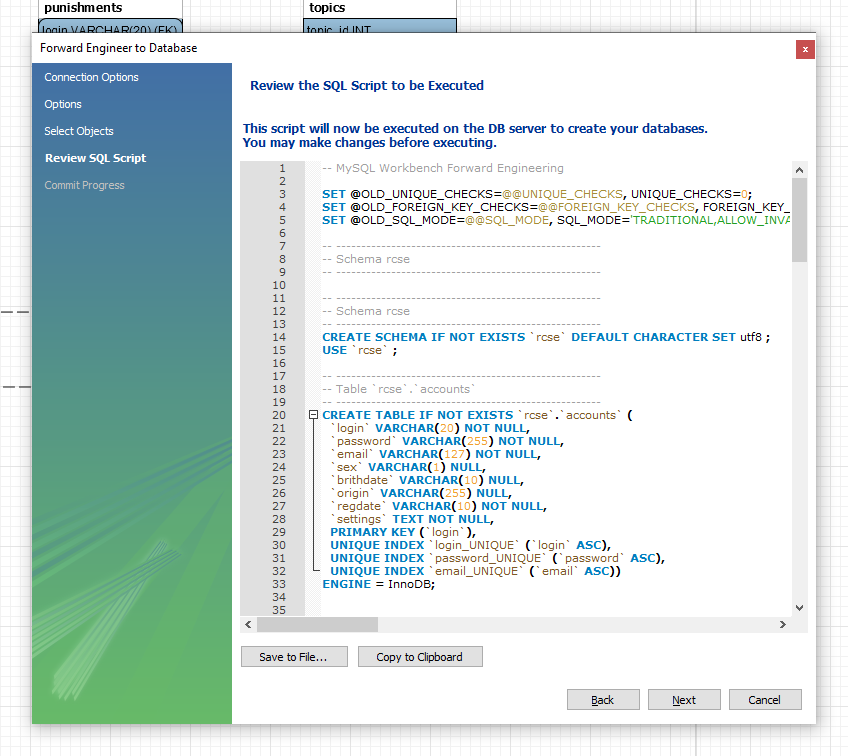


Рисунок 2.20 – Окно загрузки базы данных

После сообщения от программы об успешной загрузке, стоит зайти в PhpMyAdmin и убедиться, что БД загружена правильно. Экран PMA с загруженной базой данных изображён на рисунке 2.21.

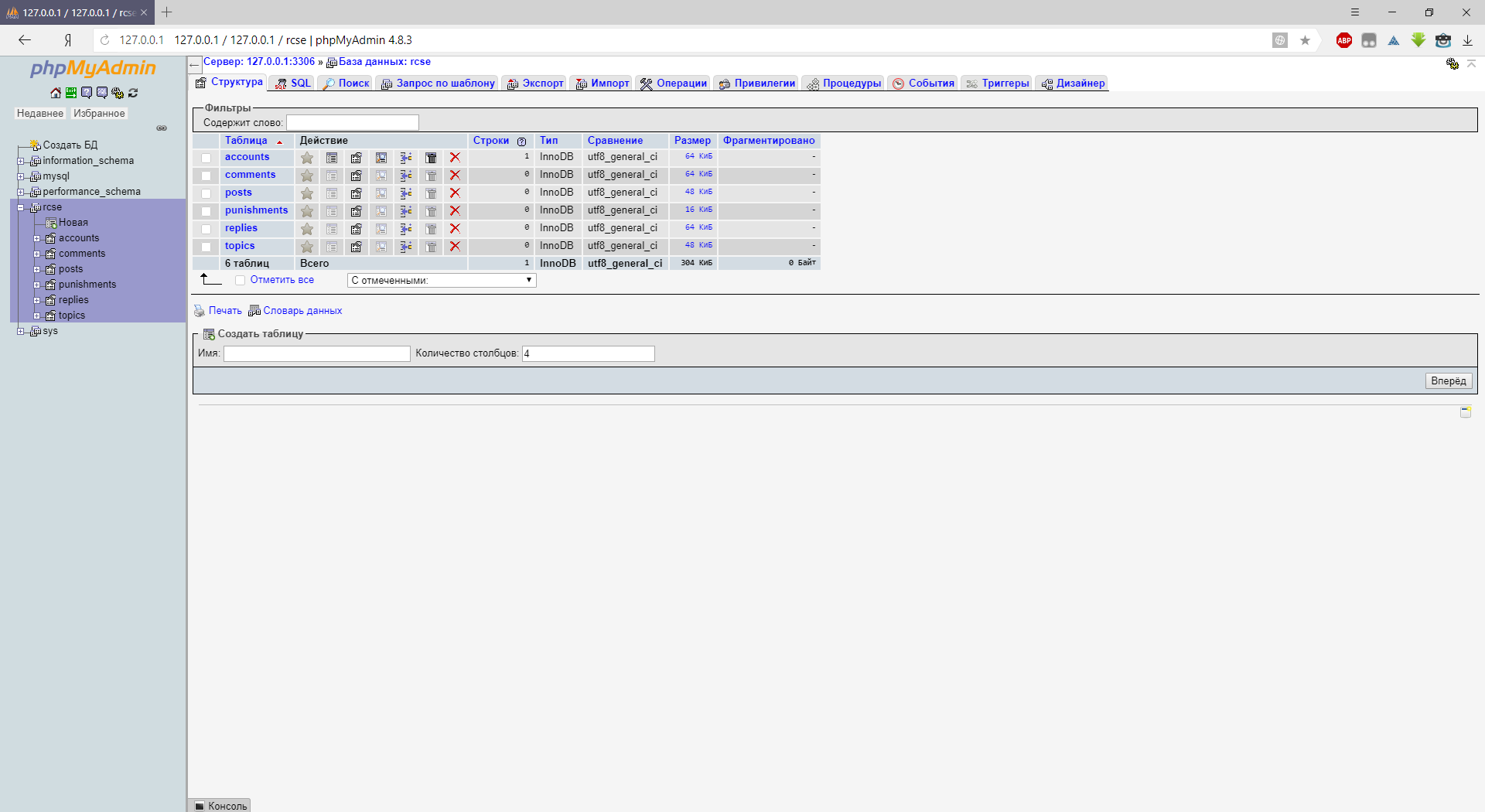


Рисунок 2.21 – Загруженная БД в обозревателе PhpMyAdmin

Далее ведётся разработка модулей движка. Для наглядности, рассмотрим DBManager. На рисунке 2.22 показана часть исходного кода этого класса.

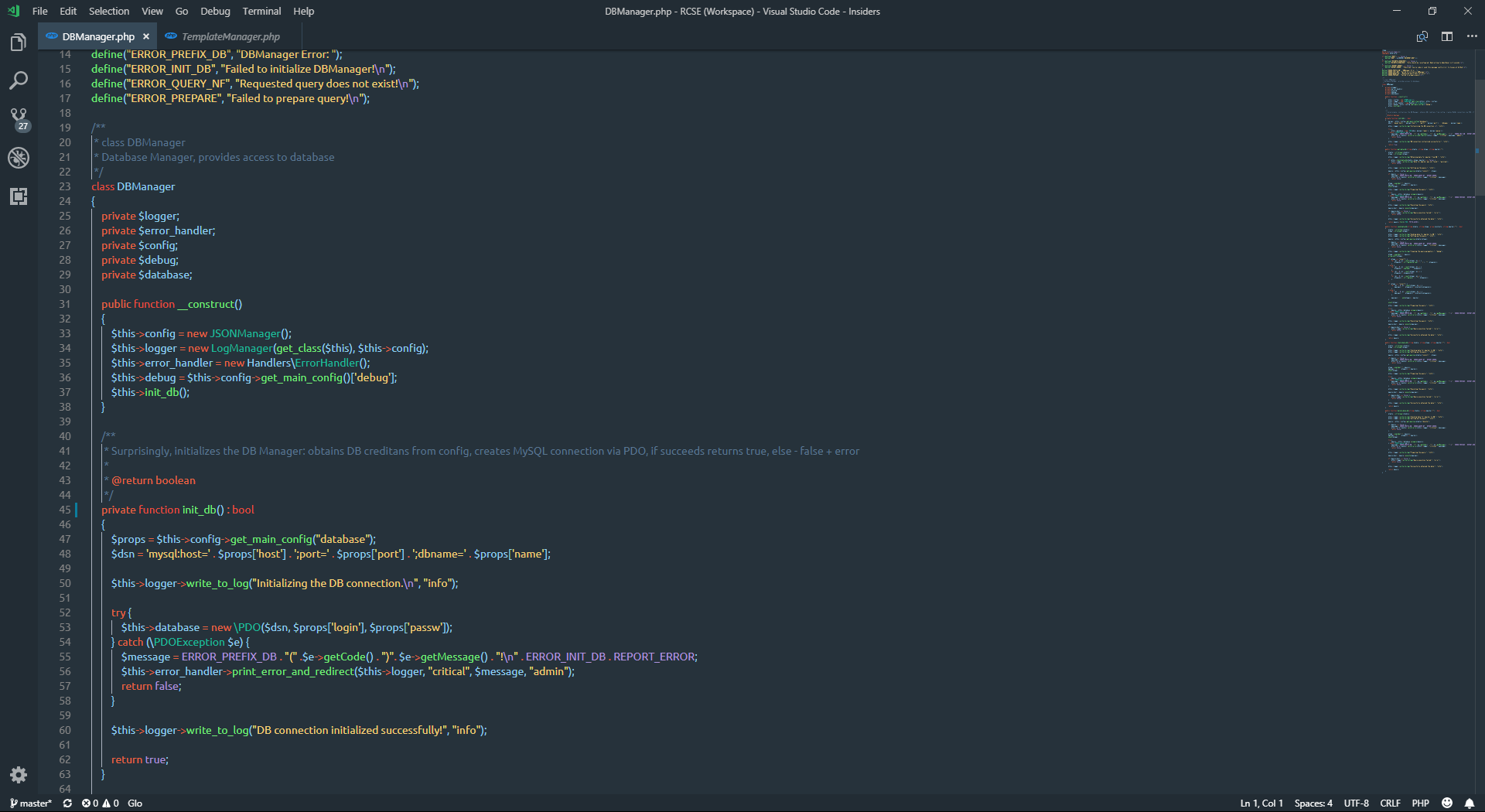


Рисунок 2.22 – Исходный код класса DBManager

Теперь необходимо убедиться в работоспособности разработанного модуля. Для этого нужно сделать простой тестовый файл, код которого изображён на рисунке 2.23.

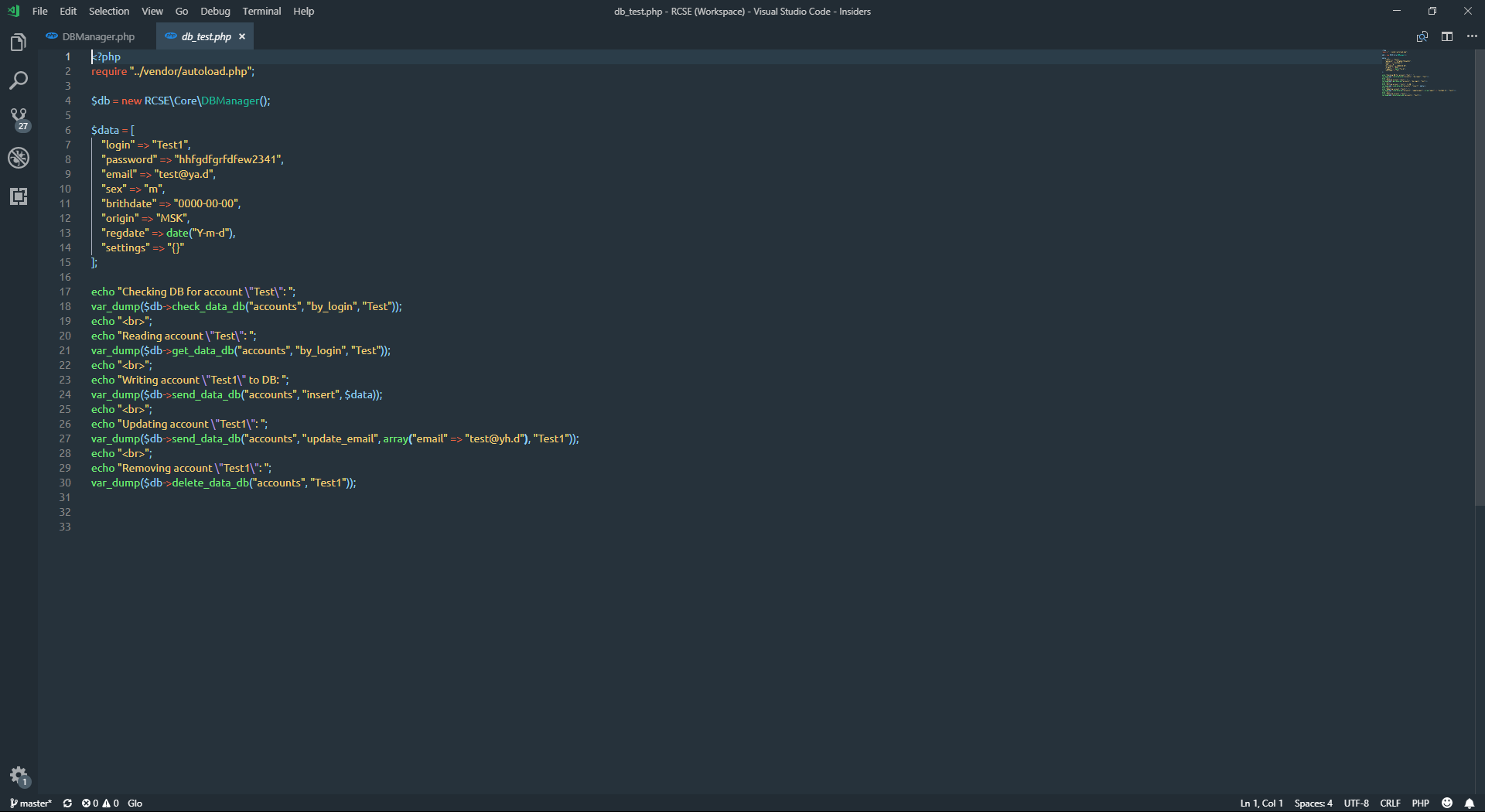


Рисунок 2.23 – Исходный код тестового файла

Результат исполнения этого файла показан на рисунке 2.24.

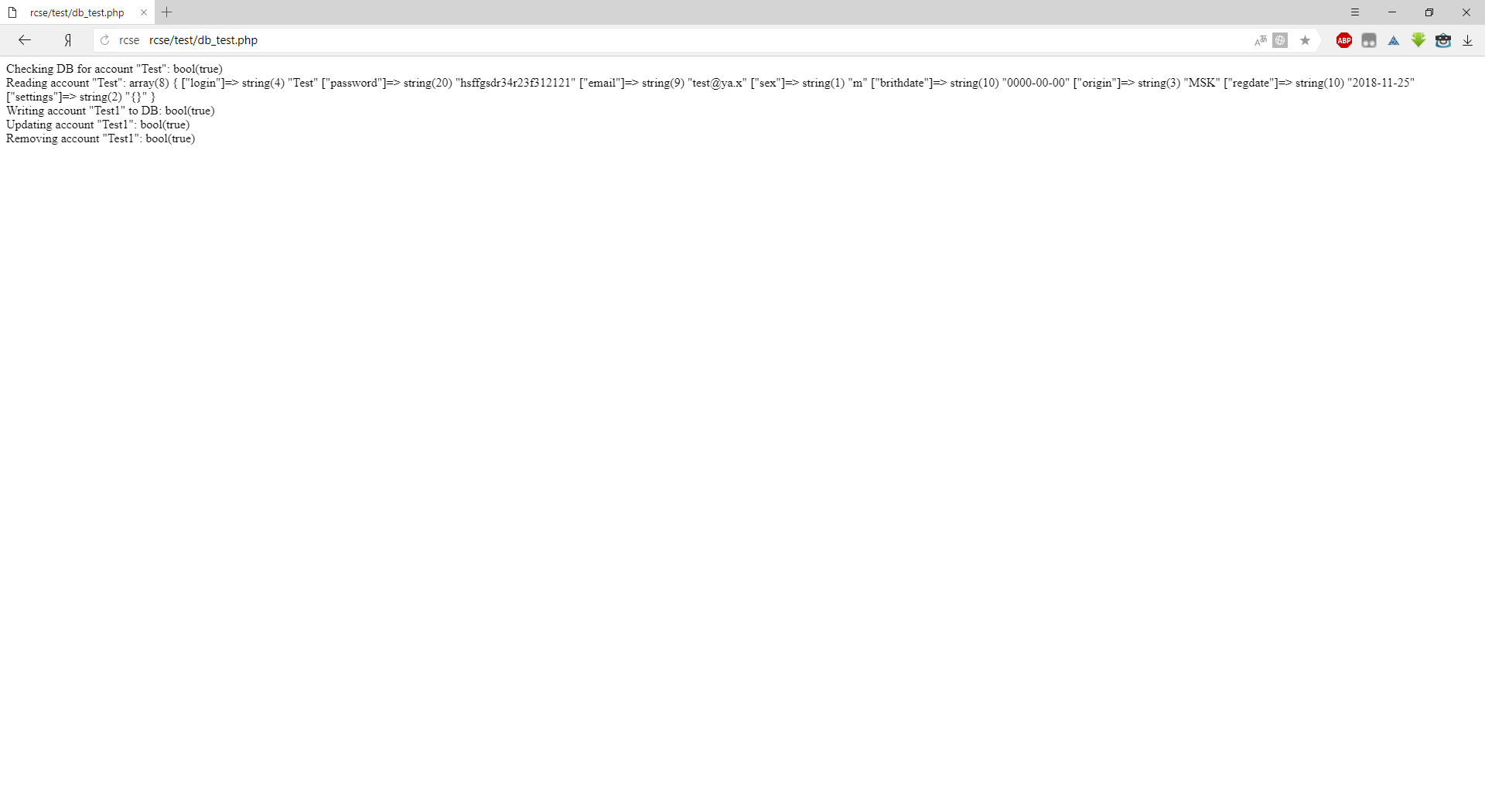


Рисунок 2.24 – Результат исполнения тестового файла

2.6 Разработка макета сайта

Теперь нужно разработать шаблон страницы по созданному макету, который будет загружаться при открытии сайта. Код страницы в HTML и CSS представлен на рисунке 2.25.

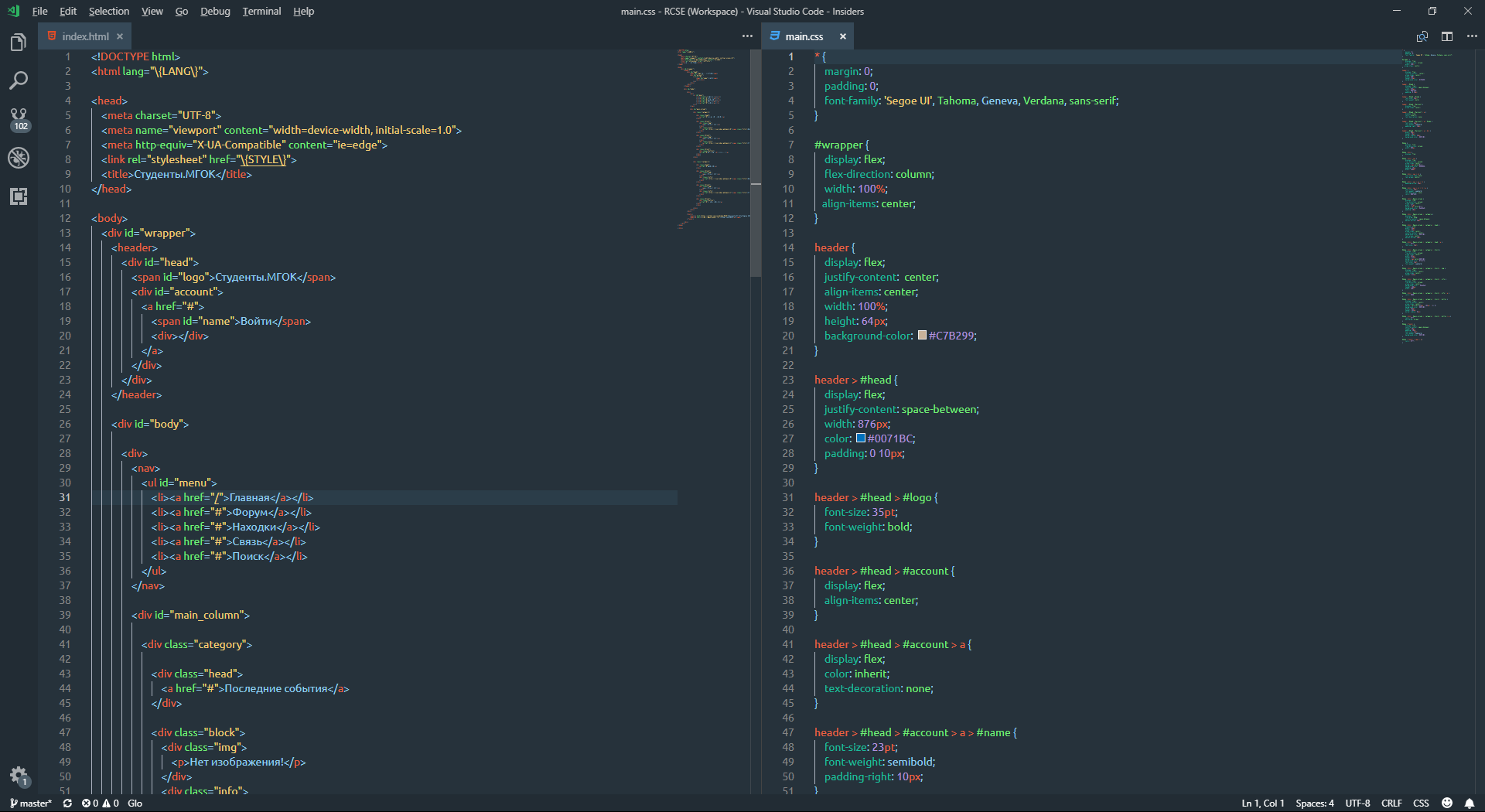


Рисунок 2.25 – Исходные коды первого шаблона

Теперь дело за малым – научить движок загружать этот шаблон! Для этого потребуется создать index.php в корневом каталоге и ввести в него код, указанный на рисунке 2.26.

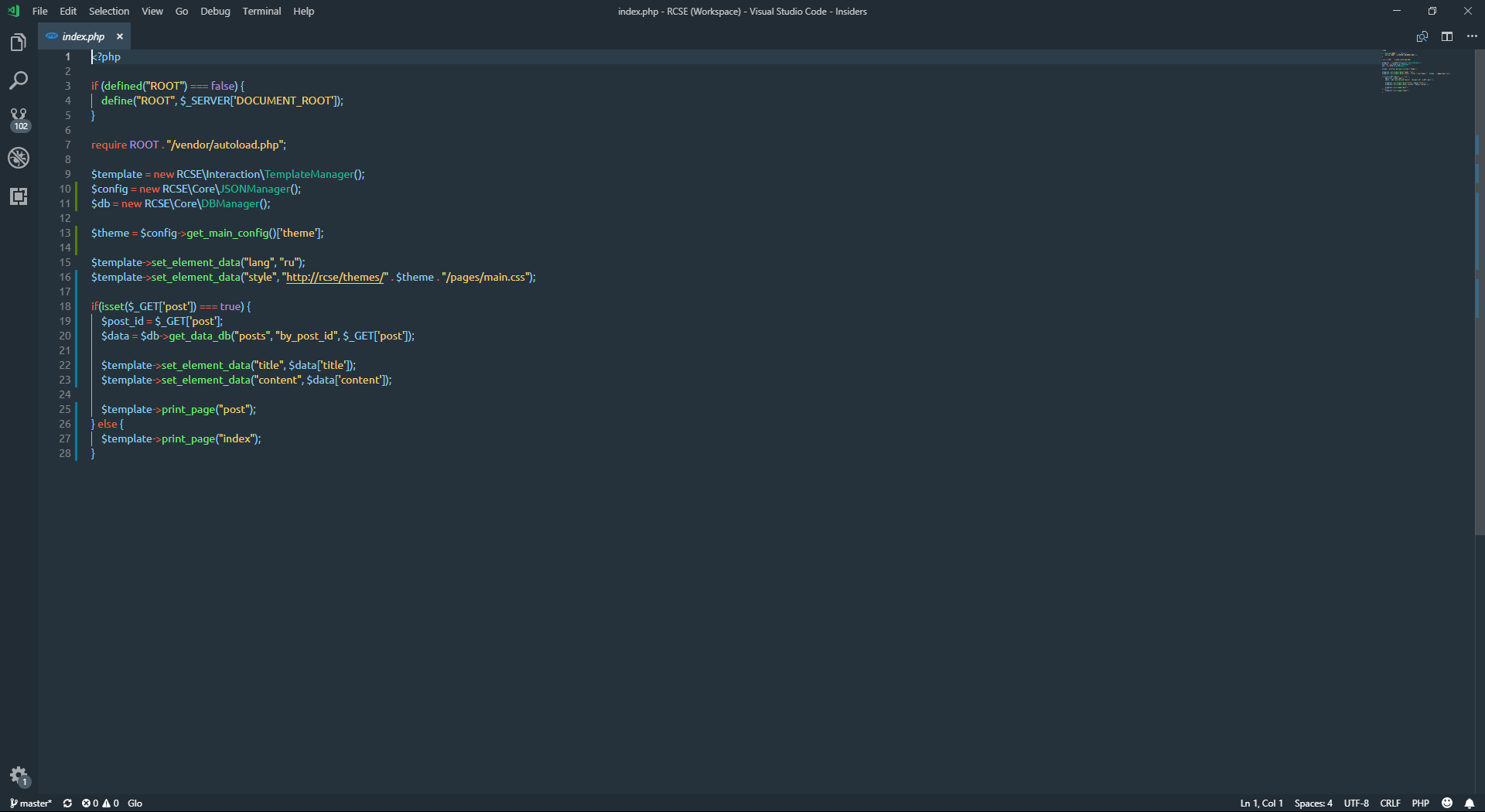


Рисунок 2.26 – Код файла index.php

Открываем сайт в браузере и видим страницу, изображённую на рисунке 2.27.

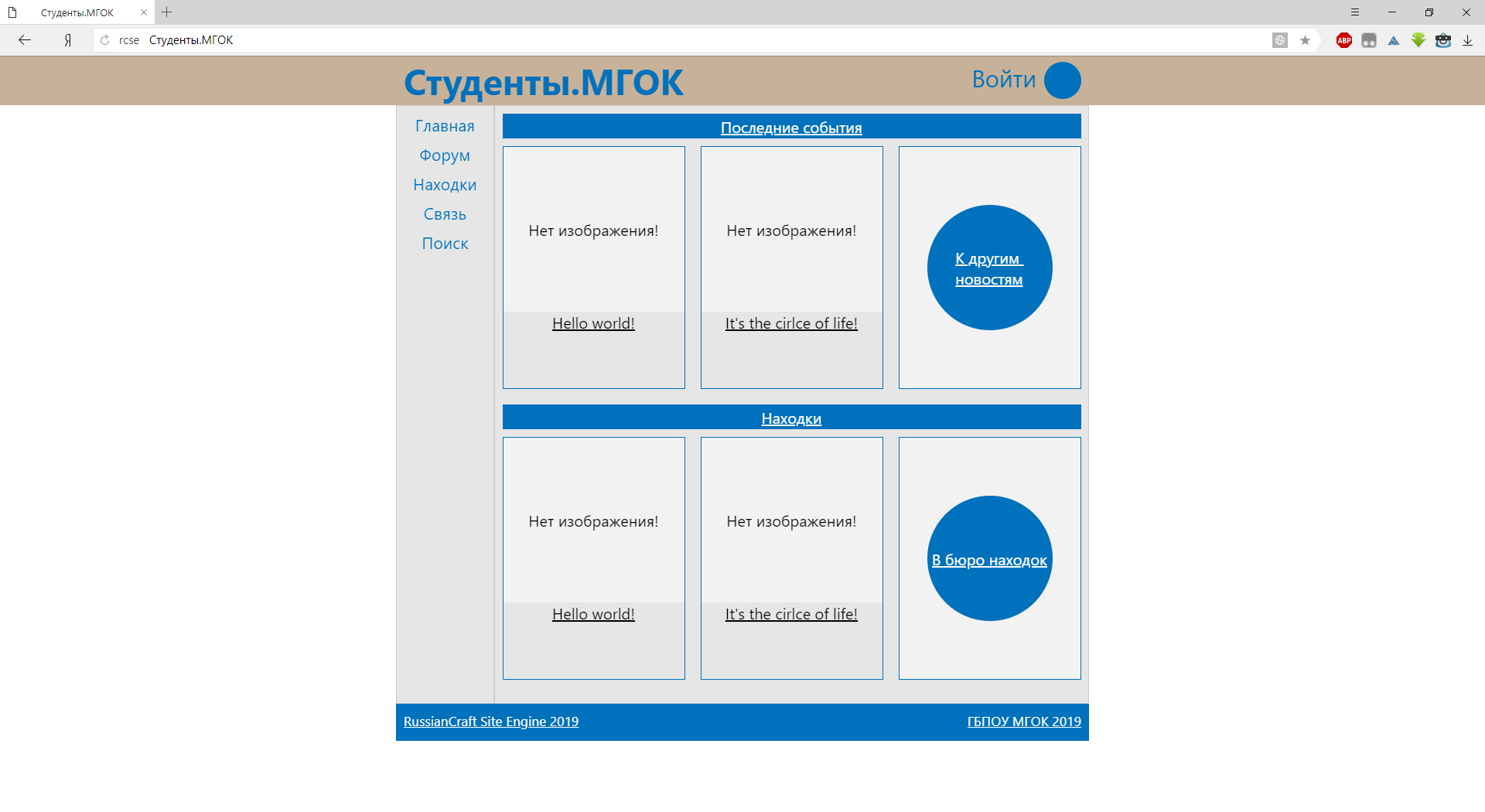


Рисунок 2.27 – Главная страница через веб-сервер

ЗАКЛЮЧЕНИЕ

Современный Интернет насыщен разнообразной информацией, и число компаний и частных лиц, желающих делиться своей растёт. Можно создавать сайты вручную, тратя на это порой годы, уходя из-за затянувшегося процесса разработки в депрессию и забрасывая, возможно, лучшую идею в сети.

Здесь на помощь приходят CMS. Они позволяют легко создать сайт, наполнить его информацией и начать продвигать его. И рынок CMS систем с каждым годом только растёт – множество молодых специалистов желают потягаться с ветеранами индустрии. И кто знает, возможно однажды кому-нибудь удастся обойти WordPress в списке лидеров. И, при удачном стечении обстоятельств, возможно это будет даже RCSE, созданный в рамках этого курсового проекта.

Но даже если успех не придёт – эта работа принесла немало новых знаний и опыта, которые помогут в новых начинаниях, и однажды выльются во что-то стоящее.

Согласно техническому заданию, целью этого курсового проекта было создание системы управления контентом RussianCraft Site Engine и запуск на ней сайта «Студенты.МГОК».

В процессе выполнения этой работы был проанализирован рынок CMS систем, отточены навыки программирования на PHP, создания страниц на HTML\CSS и работы с базами данных MySQL. Были изучены новые способы взаимодействия элементов веб-приложения друг с другом. Сайт работает в тестовом режиме и пополняется функциями с каждым обновлением.

Таким образом, можно констатировать, что цель работы достигнута, а задачи решены.

СПИСОК ИСПОЛЬЗОВАНЫХ ИСТОЧНИКОВ

1. Система управления контентом – Википедия [Электрон. ресурс] – Электрон. данные – Режим доступа: https://ru.wikipedia.org/wiki/Система\_управления\_содержимым
2. Usage Statistics and Market Share of Content Management Systems for Websites, December 2018 [Электрон. ресурс] – Электрон. данные – Режим доступа: https://w3techs.com/technologies/overview/content\_management/all
3. PHP – Википедия [Электрон. ресурс] – Электронные данные – Режим доступа: https://ru.wikipedia.org/wiki/PHP
4. HTML – Википедия [Электрон. ресурс] – Электронные данные – Режим доступа: https://ru.wikipedia.org/wiki/HTML
5. CSS – Википедия [Электрон. ресурс] – Электронные данные – Режим доступа: https://ru.wikipedia.org/wiki/CSS
6. Visual Studio Code – Википедия [Электрон. ресурс] – Электронные данные – Режим доступа: https://ru.wikipedia.org/wiki/Visual\_Studio\_Code
7. MySQL Workbench – Википедия [Электрон. ресурс] – Электронные данные – Режим доступа: https://ru.wikipedia.org/wiki/MySQL\_Workbench

ПРИЛОЖЕНИЕ А

(обязательное)

Назначение CMS

Основным назначением разработки является создание системы управления конфетном.

Целью создания CMS является: создание системы управления контентом «RussianCraft Site Engine».

Целевой аудиторией продукта являются веб-мастера, системные администраторы и блогеры.

CMS должна обеспечивать выполнение следующих задач:

* предоставлять инструментарий для создания сайтов;
* предоставлять возможность гибко изменять функционал системы для достижения задач сайта, запускаемого на ней;
* предоставить стабильный продукт, качественно защищающий личные данные пользователей сервиса.

Первичная разработка и запуск сайта на базе CMS должен производится силами Исполнителя при непосредственном участии Заказчика. Заказчик предоставляет необходимые Исполнителю текстовые и графические материалы, а также комментарии, касающиеся их содержания, объёма, оформления и размещения.

Запускаемый на системе сайт должен иметь русскоязычный интерфейс, все страницы сайта должны иметь корректные ссылки.

Система управления контентом должна предоставить администратору возможность выполнения следующих действий:

* поддержание нормального функционирования сайта;
* принятие сообщений об ошибках или нарушениях правил из обратной связи;
* редактирование и создание страниц.

Изменение дизайна и структуры сайта, а также доработка существующего и создание нового функционала должны происходить в рамках процедур поддержки сайта Исполнителем либо в соответствии с отдельными договорами на указанные виды работ. Компоновка страниц сайта должна обеспечивать автоматическое масштабирование страниц.

А.1 Структура приложения

Общая структура CMS представлена в соответствии с рисунком А.2.

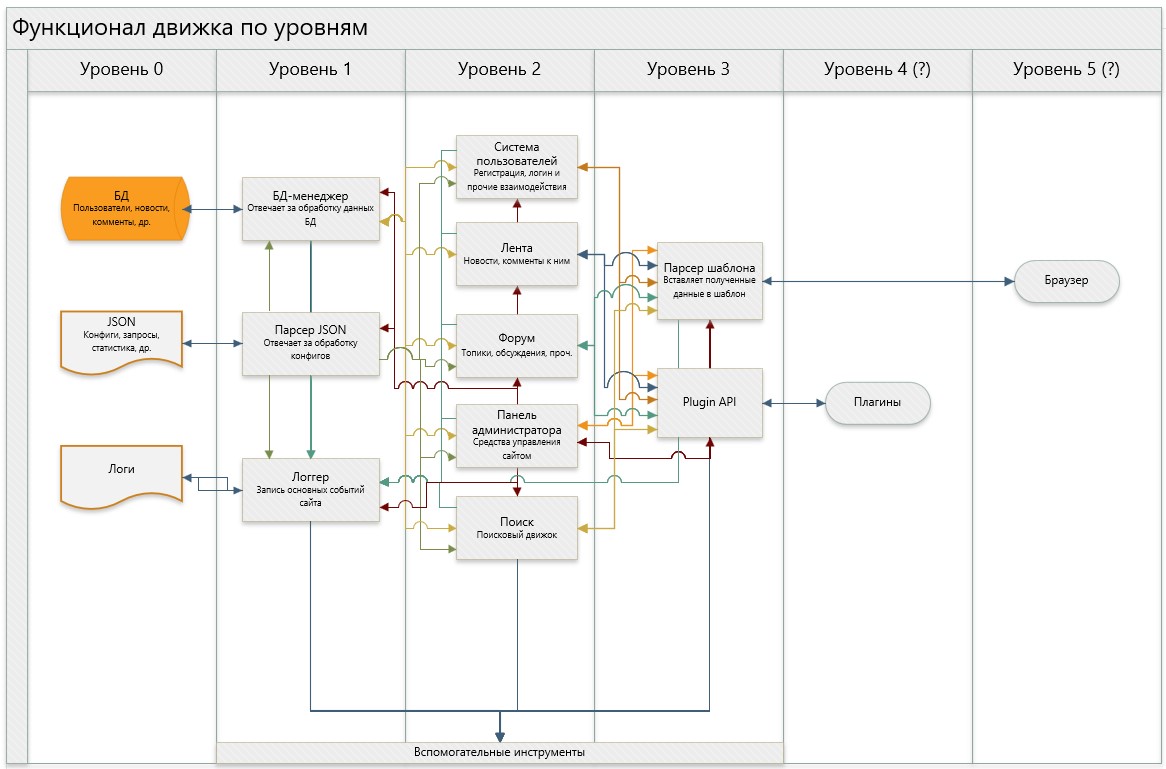


Рисунок А.2 – Общая структура CMS

Ниже представлена структура CMS в виде списка:

* Парсер шаблона;
* Plugin API;
* Система пользователей;
* Лента;
* Форум;
* Панель администрирования;
* Поиск.

Парсер шаблона является основной точкой входа на Приложение.

Назначением парсера шаблона является обеспечение доступа к данным сайта в удобочитаемом графическом виде.

ПРИЛОЖЕНИЕ Б

(рекомендуемое)

Код программного продукта

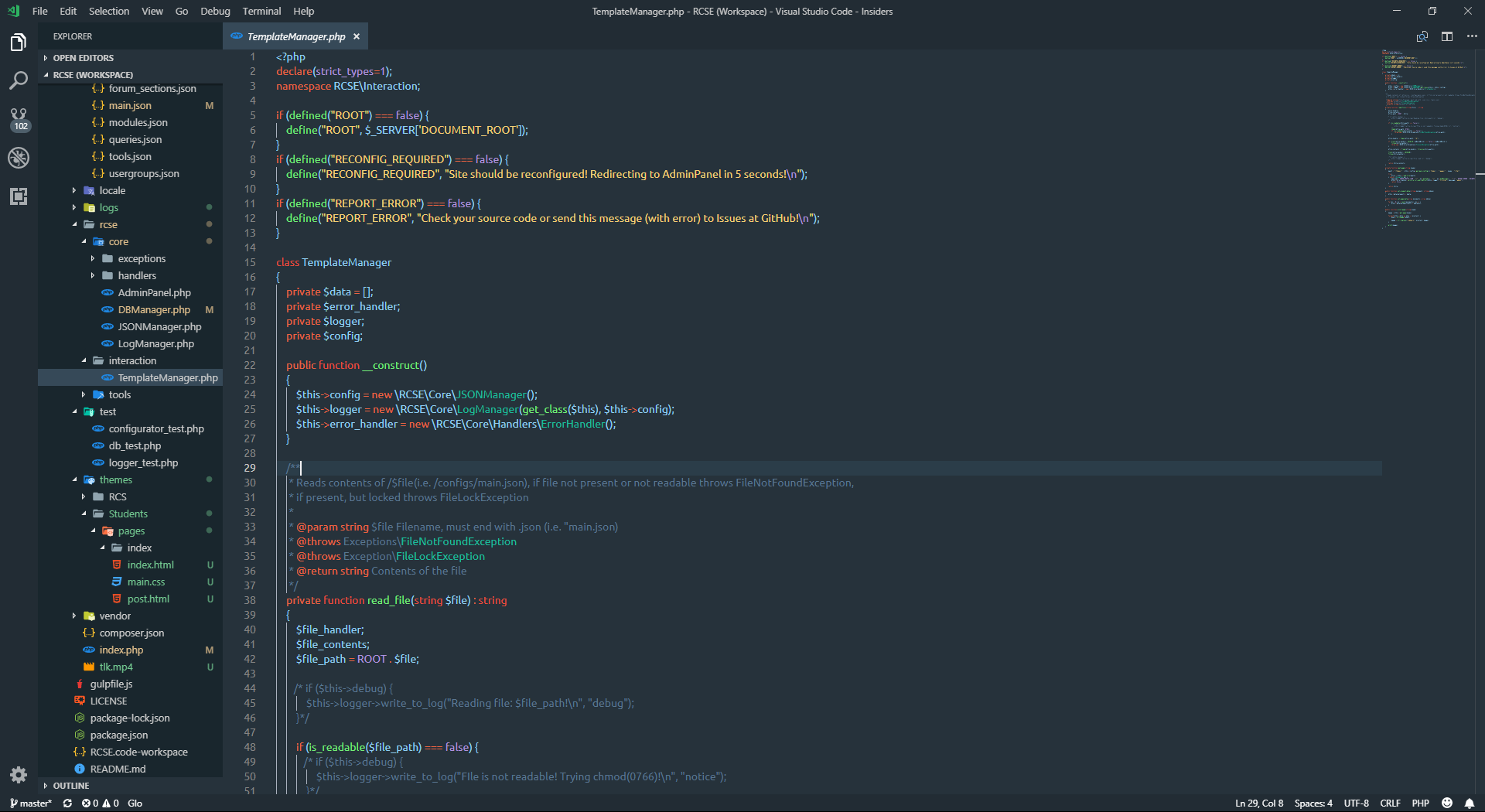


Рисунок Б.1 – Программный код парсера шаблона

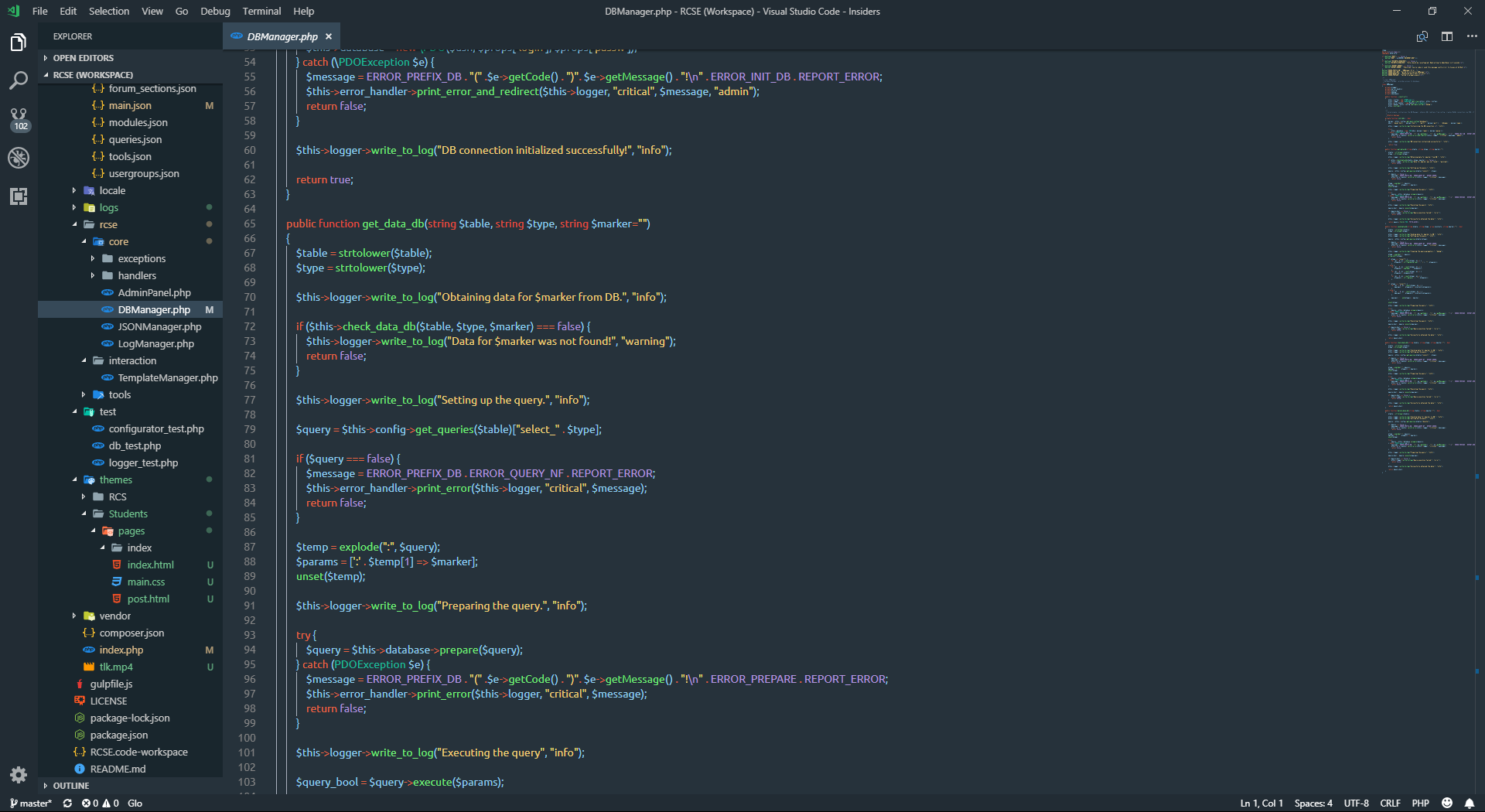


Рисунок Б.2 – Программный код обработчика базы данных

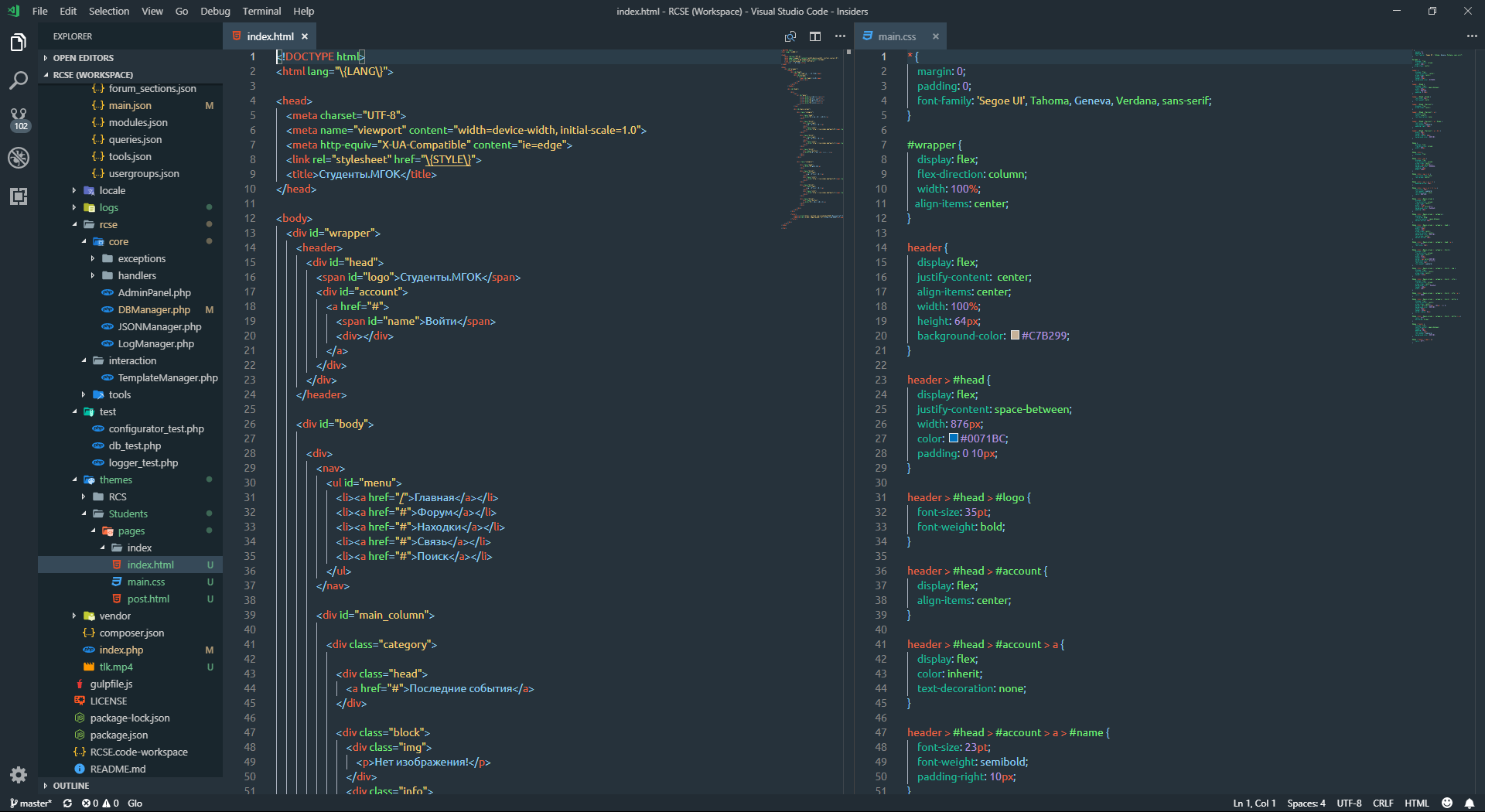


Рисунок Б.3 – Код шаблона страницы тестового сайта

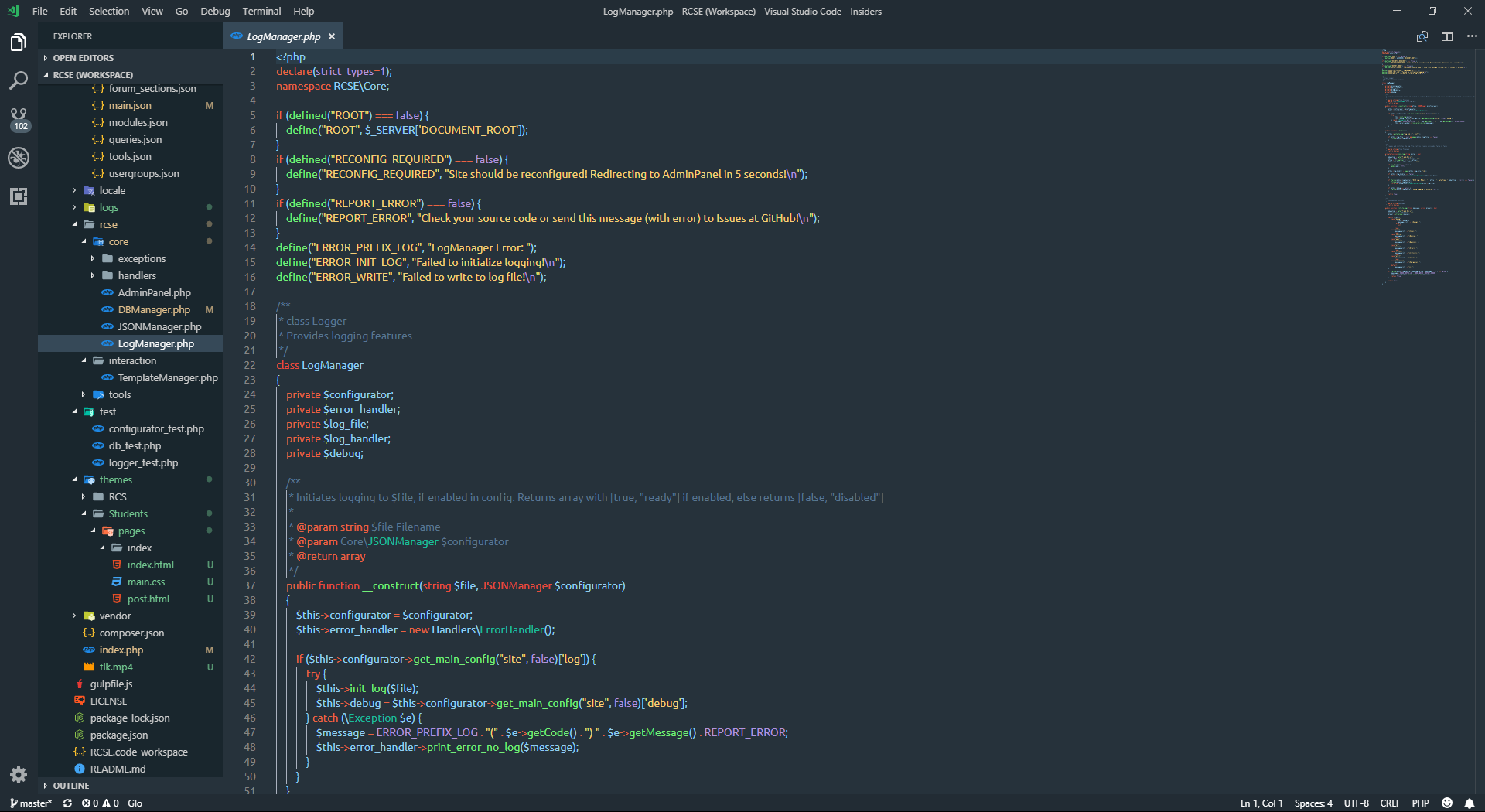


Рисунок Б.4 – Код логгера

ПРИЛОЖЕНИЕ В

(рекомендуемое)

Программный продукт

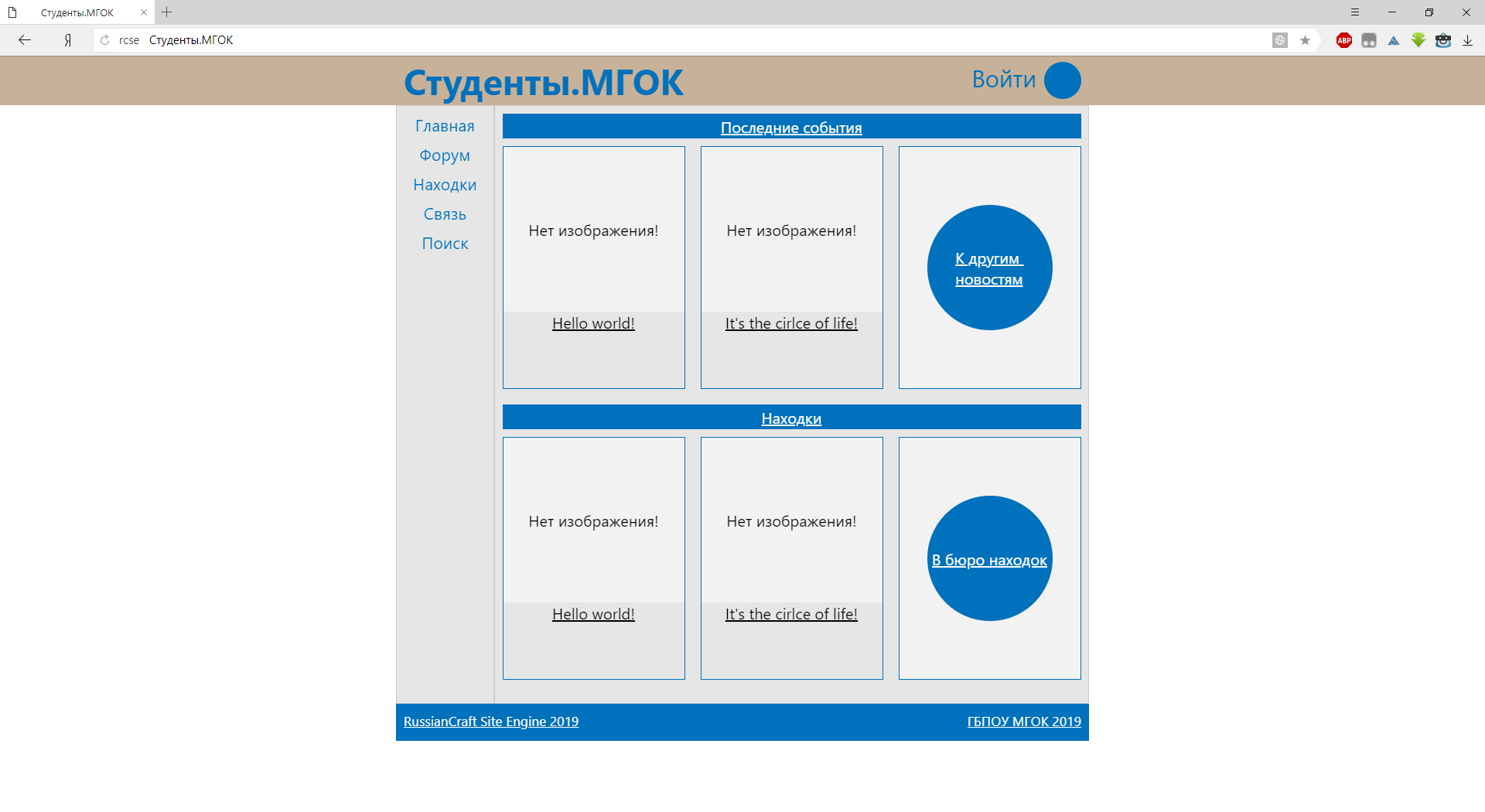


Рисунок В.1 – Главная страница тестового сайта

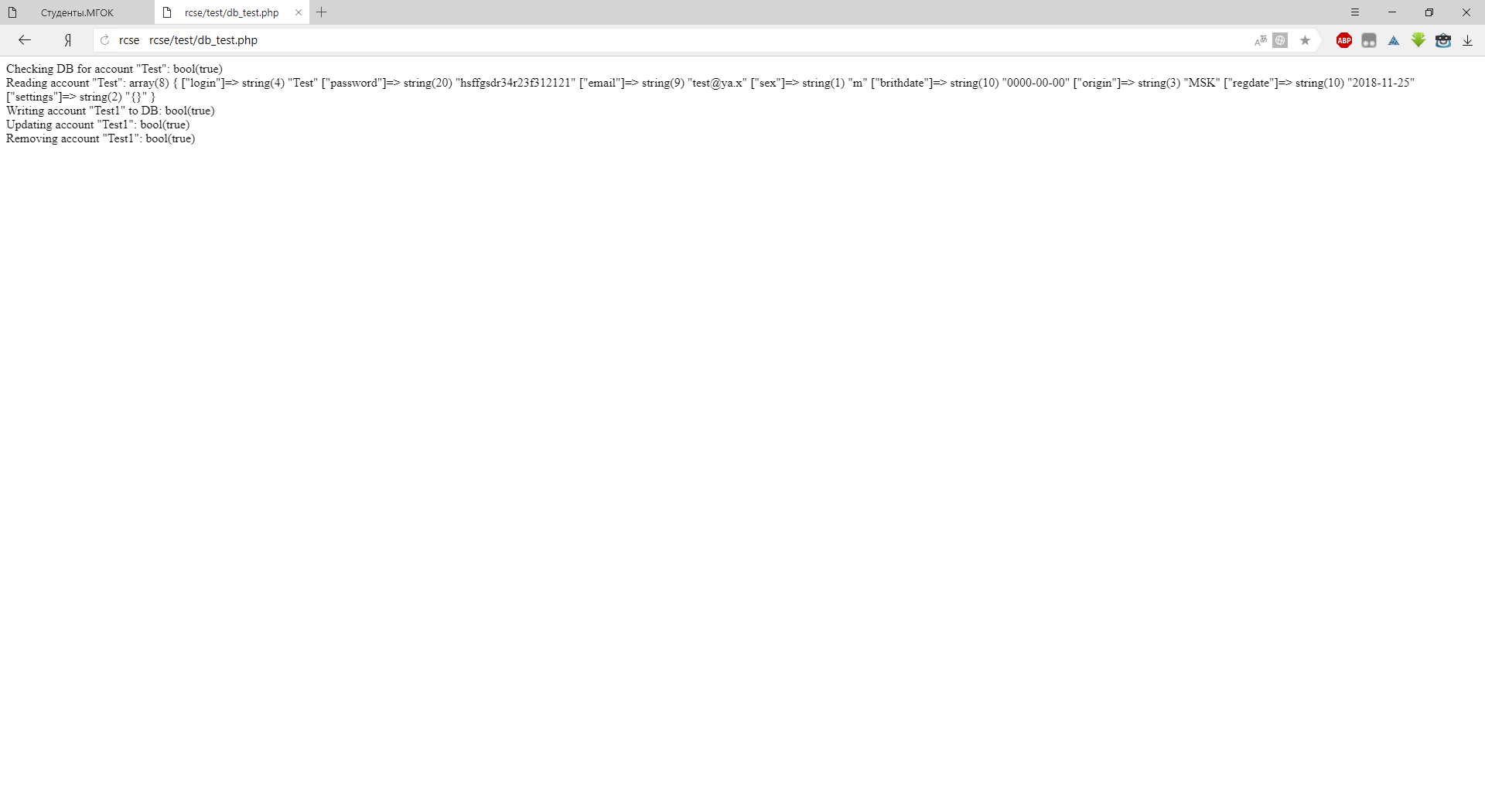


Рисунок В.2 – Тест обработчика базы данных