

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий

институт

Вычислительная техника

кафедра

УТВЕРЖДАЮ

Заведующий кафедрой

О. В. Непомнящий

подпись

инициалы, фамилия

« ____ » _____

2020г.

БАКАЛАВРСКАЯ РАБОТА

09.03.01 – «Информатика и вычислительная техника»

код - наименование направления

Веб-сайт для инвестирования и мониторинга фондового рынка

тема

Руководитель

канд. техн. наук, доцент

М. С. Медведев

инициалы, фамилия

Выпускник

Д. И. Козлов

инициалы, фамилия

Нормоконтролер

канд. техн. наук, доцент

М. С. Медведев

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий

институт

Вычислительная техника

кафедра

УТВЕРЖДАЮ

Заведующий кафедрой

О. В. Непомнящий

подпись

инициалы, фамилия

« ____ » _____ 2020г.

**ЗАДАНИЕ
НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ**

в форме _____

бакалаврской работы

бакалаврской работы, дипломного проекта, дипломной работы, магистерской диссертации

Студенту _____ Козлов Дмитрий Игоревич _____

фамилия, имя, отчество

Группа ВКИ15-06Б Направление (специальность) _____ 09.03.01

Номер

Код

Информатика и вычислительная техника

наименование

Тема выпускной квалификационной работы _____ Веб-сайт для инвестирования
и мониторинга рынка _____

Утверждена приказом по университету № _____ от _____

Руководитель ВКР _____ М. С. Медведев , канд. техн. наук, доцент, ИКИТ СФУ

инициалы, фамилия, должность, ученое звание и место работы

Исходные данные для ВКР _____ Задание на ВКР _____

Перечень разделов ВКР _____ Задание на ВКР, анализ задания, проектирование, описание
принципа работы сайта, заключение. _____

Перечень графического материала _____ Презентация, выполненная с помощью
Microsoft PowerPoint 2010. _____

Руководитель ВКР _____ М. С. Медведев _____

подпись

инициалы и фамилия

Задание принял к исполнению _____ Д. И. Козлов _____

подпись, инициалы и фамилия студента

« _____ » _____ 2020г.

РЕФЕРАТ

Выпускная квалификационная работа по теме: «Веб-сайт для инвестирования и мониторинга фондового рынка». Пояснительная записка содержит 54 страниц текстового документа, 19 иллюстраций, 2 приложения, и 13 использованных источников.

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ, Open Server, Visual Studio Code, Laravel, phpMyAdmin, браузер.

Цель работы: Создать сайт для конкретного паевого фонда, с простым и удобным интерфейсом, личным кабинетом, возможностью пополнять личный счет. Сайт должен быть информативным, с возможностью создавать и размещать новости рынка, новости фонда.

В результате выполнения ВКР было разработан сайт, полностью соответствующий поставленным задачам.

В выпускную квалификационную работу входит введение, 3 главы и заключение.

Во введении ставится цель и производится ее декомпозиция на задачи.

В первой главе анализируется и описываются средства для выполнения поставленных задач, определяем платформу для дальнейшей работы.

Во второй главе описывается проектирование сайта, разрабатывается структура сайта, схема БД, идет создание и описание контроллеров, моделей.

В третьей главе описывается реализованный сайт, объясняется работа элементов.

В заключении подводятся итоги по выполненной работе.

Задание для разработки.

Задачи, которые необходимо выполнить для реализации поставленной цели:

- анализ и выбор CMS или фреймворка;
- разработка шаблона и дизайна сайта;
- разработка схемы сайта;
- подключение необходимой среды разработки;
- верстка разделов сайта;
- подключение и добавление элементов JS , PHP, подключение баз данных

MySQL;

- разработка личного кабинета.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	5
1 Анализ технического задания	6
1.1 Выбор CMS	6
1.1.1 Система управления «WordPress»	7
1.1.2 Система управления «Joomla»	8
1.1.3 Система управления «1С Битрикс»	8
1.1.4 Система управления «Drupal»	9
1.1.5 Фреймворки	10
1.1.6 Вывод по разделу	11
1.2 Определение программных средств разработки.	11
1.2.1 Язык гипертекстовой разметки «HTML»	12
1.2.2 Язык каскадных таблиц стилей «CSS»	12
1.2.3 Язык программирования «JavaScript(jQuery)»	13
1.2.4 Язык программирования «PHP»	13
1.2.5 Базы данных «MySQL»	14
2. Проектирование	16
2.1 Подготовка	16
2.2 Общая схема работы сайта	19
2.2.1 Модели	19
2.2.2 Контроллеры	21
2.2.3 Маршрутизация	22
2.2.4 Blade шаблоны	23
3. Разработка сайта и описание сайта	27
3.1 Главная страниц	27
3.2 Header-основное меню сайта	28
3.4 Новости	28
3.5 Отчет фонда	30
3.6 Контакты	31
3.7 Регистрация и аутентификация	33
3.8 Личный кабинет	35
4 Заключение	38
СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ	39

ВВЕДЕНИЕ

Паевой инвестиционный фонд (ПИФ) - фонд, основанный на доверительном управлении консолидированными активами вкладчиков специализированной управляющей компанией. Простой и надёжный финансовый инструмент, как для профессионалов, так и для неопытных инвесторов. Вкладывая средства в паевой инвестиционный фонд, инвестор может рассчитывать на получение большей доходности, чем по банковским вкладам.

Актуальность задачи в создании простого в использовании финансового инструмента не вызывает сомнения, а вкладчик с легкостью сможет отслеживать рынки и сделки «ПИФ» без глубоких познаний финансовых рынков.

Подводя итоги, мы получаем цель, создать финансовый инструмент в виде простого сайта, на котором можно будет следить за рынком. Разделы, где можно будет отслеживать экономические новости, отслеживать действия фонда, успехи и неудачи паевого общества. Личный кабинет клиента с возможностью инвестирования в «ПИФ» и отслеживания своей прибыли.

Создание сайтов сегодня может производиться разными способами, от написания на чистом коде, до использования фреймворков, конструкторов сайтов или CMS. У каждого имеются собственные особенности, недостатки и позитивные свойства, поэтому при решении вопроса, на чем лучше создавать собственный сайт, необходимо обладать достаточной информацией о доступных вариантах.

1 Анализ технического задания

1.1 Выбор CMS

CMS (Content management system) — это конструктор, позволяющий пользователю, не владеющему программированием, создавать динамические сайты, заниматься их наполнением и управлять содержимым сайта.

Главные функции CMS:

- предоставление инструментов для создания содержимого, организация совместной работы над содержимым;
- управление содержимым: хранение, контроль версий, соблюдение режима доступа, управление потоком документов и т. п.;
- публикация содержимого;
- представление информации в виде, удобном для навигации, поиска.

В системе управления содержимым могут находиться самые различные данные: документы, фильмы, фотографии, номера телефонов, научные данные и так далее. Такая система часто используется для хранения, управления, пересмотра и публикации документации.

Существует несколько основных CMS:

- WordPress – известный инструмент, который изначально предназначался для создания блогов, однако на данный момент имеет куда более широкую функциональность;
- Joomla – крайне простая в использовании система, которую применяют начинающие разработчики, не требующие от ресурса мощных вычислений;
- 1С Битрикс – объемная, многопрофильная платная система, чересчур тяжеловесная для простых задач, но хорошо справляющаяся со сложными;
- MODx – удобная для разработчиков бесплатная CMS, обладающая высокой степенью защищенности и достаточной гибкостью для решения большинства задач;

- Drupal – бесплатная, но полнофункциональная и достаточно тяжелая CMS, имеющая в составе все необходимое для создания полноценного сайта;

- DLE – отчасти аналог предыдущей системы, простой в использовании и интуитивно понятный;

- движки для создания форумов: phpBB, vBulletin и другие;

- системы для организации интернет-магазинов: как бесплатные (OpenCart, PrestaShop), так и платные (Umi.CMS, Shop-Script и другие);

1.1.1 Система управления «WordPress»

WordPress — самая популярная CMS в мире, на которой работает больше 60% сайтов. Распространяется по свободной лицензии и предлагает невероятное количество шаблонов, плагинов и обучающих материалов.

CMS WordPress изначально была платформой для запуска блогов, но удачная архитектура и удобная система расширения функциональности за счёт установки плагинов сделали движок универсальным. На основе WP созданы сайты крупнейших газет, международных брендов, стран.

Плюсы и минусы:

- простая, но в то же время требующая углубленного изучения панель администратора;

- множество тем имеют премиальный статус, что подразумевает под собой не малую плату;

- достаточно весома нагрузка на сервер;

- порою низкая скорость загрузки сайта;

- ограниченный набор базовых функций;

+ встроенный файловый менеджер, которым стало пользоваться ещё удобнее благодаря внедрению редактора Gutenberg;

+ обширная библиотека плагинов, расширяющих базовую функциональность движка;

+ компетентное сообщество и большое количество обучающих материалов.

1.1.2 Система управления «Joomla»

Joomla — CMS , предоставляющий пользователям достаточно возможностей для создания сайтов разных типов. Система подходит начинающим вебмастерам, которые предпочитают пользоваться готовыми решениями, а также профессиональным разработчикам.

Важно отметить, что Joomla представляет собой скрипты из PHP и JavaScript. Это распространенные языки программирования. Кроме того, продукт имеет открытый исходный код. При наличии специальных знаний можно вносить свои изменения в скрипт, осуществлять тонкую настройку и повышать функционал.

Плюсы и минусы:

- + полностью бесплатная лицензия;
- + открытый исходный код;
- + низкие требования к серверу;
- + удобное редактирование для контента;
- наличие уязвимостей, проекты под управлением этой управляющей системы часто взламывают;
- возникающие проблемы с индексацией сайта;
- отсутствие системы поддержки.

1.1.3 Система управления «1С Битрикс»

1С-Битрикс — коммерческая CMS, которая предлагает готовые решения для создания разных типов сайтов. В первую очередь подходит для запуска проектов с высокой посещаемостью и серьезными требованиями безопасности. Например, на ней работают интернет-магазины некоторых федеральных сетей техники.

В российском интернет-пространстве есть устойчивый стереотип, который превращает Битрикс в CMS для интернет-магазина. На самом деле продукт

подходит для разных целей. На базе него можно создать портфолио, информационный сайт, файловое хранилище, корпоративный портал или онлайн-сообщество.

Без сомнений, систему управления сайтом 1С-Битрикс можно считать одной из лучших CMS, не имеющих себе равных. Но стоит заметить, что ее использование применимо только для крупных проектов. Система требовательна к ресурсам, вплоть до выделения отдельного сервера, и к профессиональным навыкам разработчиков.

Плюсы и минусы:

- + широкие возможности;
- + надежная защита;
- + неограниченный потенциал масштабирования;
- + библиотека плагинов;
- За обновления необходимо платить каждый год;
- Высокие требования и нагрузка к серверам;
- Высокая сложность написания сложных проектов.

1.1.4 Система управления «Drupal»

Drupal — это абсолютно бесплатный движок, развитием которого занимается большое сообщество пользователей с высоким уровнем технической подготовки. Платформа доступна для скачивания в виде ядра системы и тематических сборок, в состав которых входят все необходимые модули для решения конкретной задачи — например, построения интернет-магазина или информационного портала с элементами соцсети.

Главная черта Drupal — гибкость. Фактически, это своего рода конструктор сайтов, из которого вебмастер создаёт систему с нужной для себя функциональностью. Особенно это касается работы с ядром системы. «Из

коробки» доступны только самые основные возможности. С помощью модулей вебмастер формирует конфигурацию, которая позволит реализовать его проект.

Плюсы и минусы:

- + открытый исходный код;
- + возможность установки тем и расширений;
- + кэширование страниц;
- + подробная статистика и контроль пользователей;
- сложный интерфейс для восприятия;
- высокие технические требования к серверу;
- отсутствие файлового менеджера;
- система управления сложена на написании и изучении.

1.1.5 Фреймворки

Фрэймворк (иногда фреймвóрк; англицизм, неологизм от framework — остов, каркас, структура) — программная платформа, определяющая структуру программной системы; программное обеспечение, облегчающее разработку и объединение разных компонентов большого программного проекта.

Это программная среда специального назначения, своеобразный каркас, используемый для того, чтобы существенно облегчить процесс объединения определенных компонентов при создании программ. Это основа, которая позволяет добавлять компоненты в зависимости от потребностей. База, на которой можно сформировать программу любого назначения достаточно быстро и без особых затруднений.

Yii - достаточно "возрастной" фреймворк, который продолжает обновляться в наши дни. Отличается удобным функционалом - кэширование, высокая производительность, полная обработка ошибок, возможность переноса (миграции) существующих баз данных, использование jQuery и другое

Laravel - частый лидер разнообразных опросов и рейтингов, посвященных

php-фреймворкам. Проект является действительно многообещающим, получил признание достаточно опытных специалистов. Фреймворк просто освоить, является идеальным вариантом для небольших, а также средних по сложности проектов. Подойдет для быстрого, удобного написания требуемого кода.

Phalcon PHP - отличается открытым кодом (языки программирования C, C++, PHP), поддержкой практически всех современных ОС. Производительность этого фреймворка находится на высоком уровне - это подтверждено множеством специализированных тестирований, и, как следствие, его популярностью.

1.1.6 Вывод по разделу

Проанализировав большую часть предложенных систем управления сайтами, было принято решение о создании сайта без использования готовых CMS. И воспользоваться фреймворком Laravel и его функционалом, так как сравнение плюсов и минусов CMS говорит или об излишней сложности платформы, или повышенных нагрузках на сервера, а данный фреймворк показал себя с лучшей стороны в плане надежности, гибкости и стабильности. Также разработанная система более оптимизирована и имеет низкую нагрузку на сервер. Так как Laravel имеет открытый код и позволяет установку дополнительного обеспечения для упрощения некоторых аспектов разработки сайта и его функционала.

1.2 Определение программных средств разработки.

В данном пункте рассмотрены программные средства, которые понадобятся для разработки данного сайта, а именно HTML5, CSS, JavaScript (jQuery), PHP, MySQL.

1.2.1 Язык гипертекстовой разметки «HTML»

HTML (от англ. HyperText Markup Language — «язык гипертекстовой разметки») — 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.

Язык HTML до 5-й версии определялся как приложение SGML (стандартного обобщённого языка разметки по стандарту ISO 8879). Спецификации HTML5 формулируются в терминах DOM (объектной модели документа).

1.2.2 Язык каскадных таблиц стилей «CSS»

CSS (Cascading Style Sheets — каскадные таблицы стилей) — 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, например, к SVG или XUL.

CSS используется создателями веб-страниц для задания цветов, шрифтов, стилей, расположения отдельных блоков и других аспектов представления внешнего вида этих веб-страниц. Основной целью разработки CSS являлось отделение описания логической структуры веб-страницы (которое производится с помощью HTML или других языков разметки) от описания внешнего вида этой веб-страницы (которое теперь производится с помощью формального

языка CSS). Такое разделение может увеличить доступность документа, предоставить большую гибкость и возможность управления его представлением, а также уменьшить сложность и повторяемость в структурном содержимом.

1.2.3 Язык программирования «JavaScript (jQuery)»

jQuery - набор функций JavaScript, фокусирующийся на взаимодействии JavaScript и HTML. Библиотека jQuery помогает легко получать доступ к любому элементу DOM, обращаться к атрибутам и содержимому элементов DOM, манипулировать ими. Также библиотека jQuery предоставляет удобный API для работы с AJAX. Разработка jQuery ведётся командой добровольцев на пожертвования.

Точно так же, как CSS отделяет визуализацию от структуры HTML, jQuery отделяет поведение от структуры HTML. Например, вместо прямого указания на обработчика события нажатия кнопки управление передаётся jQuery, которая идентифицирует кнопки и затем преобразует его в обработчик события клика. Такое разделение поведения и структуры также называется принципом ненавязчивого JavaScript.

Библиотека jQuery содержит функциональность, полезную для максимально широкого круга задач.

1.2.4 Язык программирования «PHP»

PHP (Hypertext Preprocessor) — «PHP: препроцессор гипертекста»; первоначально Personal Home Page Tools— «Инструменты для создания персональных веб-страниц») — скриптовый язык общего 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.

Язык и его интерпретатор (Zend Engine) разрабатываются группой энтузиастов в рамках проекта с открытым кодом[16]. Проект распространяется под собственной лицензией, несовместимой с GNU GPL.

В области веб-программирования, в частности серверной части, PHP — один из популярных сценарных языков (наряду с JSP, Perl и языками, используемыми в ASP.NET).

Популярность в области построения веб-сайтов определяется наличием большого набора встроенных средств и дополнительных модулей для разработки веб-приложений. Основные из них:

- автоматическое извлечение POST- и GET-параметров, а также переменных окружения веб-сервера в предопределённые массивы,
- взаимодействие с большим количеством различных систем управления базами данных через дополнительные модули работа с HTTP-авторизацией,
- работа с cookies и сессиями,
- работа с локальными и удалёнными файлами, сокетами,
- обработка файлов, загружаемых на сервер.

1.2.5 Базы данных «MySQL»

MySQL — свободная реляционная система управления базами данных. Разработку и поддержку MySQL осуществляет корпорация Oracle, получившая права на торговую марку вместе с поглощённой Sun Microsystems, которая ранее приобрела шведскую компанию MySQL AB. Продукт распространяется как под GNU General Public License, так и под собственной коммерческой лицензией. Помимо этого, разработчики создают функциональность по заказу лицензионных пользователей. Именно благодаря такому заказу почти в самых ранних версиях появился механизм репликации.

Гибкость СУБД MySQL обеспечивается поддержкой большого количества типов таблиц: пользователи могут выбрать как таблицы типа MyISAM, поддерживающие полнотекстовый поиск, так и таблицы InnoDB, поддерживающие транзакции на уровне отдельных записей. Более того, СУБД MySQL поставляется со специальным типом таблиц EXAMPLE, демонстрирующим принципы создания новых типов таблиц. Благодаря открытой архитектуре и GPL-лицензированию, в СУБД MySQL постоянно появляются новые типы таблиц.

2. Проектирование

2.1 Подготовка

Фреймворк – это некая среда разработки для программиста с готовыми правилами и инструментами. Фреймворк, с одной стороны, помогает и ускоряет разработку, а с другой, накладывает определенные ограничения. На фреймворках делаются проекты средней сложности с посещаемостью в миллионы

Laravel — бесплатный веб-фреймворк с открытым кодом, предназначенный для разработки с использованием архитектурной модели MVC (англ. Model View Controller — модель-представление-контроллер).

Для полноценной работы фреймворка Laravel необходимо соблюдение некоторых условий. В частности, наш сервер должен иметь:

- PHP $\geq 7.2.5$,
- расширение BCMath PHP,
- расширение ctype PHP,
- расширение Fileinfo PHP,
- расширение JSON PHP,
- расширение Mbstring PHP,
- расширение OpenSSL PHP,
- расширение PDO PHP,
- расширение Tokenizer PHP,
- расширение XML PHP.

Так же MySQL, PHP Admin- пригодится только для частичного управления БД. Laravel позволяет производить все необходимые операции с БД при помощи системы объектно-реляционного отображения Eloquent —реализация шаблона ActiveRecord в Laravel для работы с базами данных.

Установка производится с помощью консольных команд из-под управления сервера. В файлах конфигурации, производится первичная настройка фреймворка, подключение к БД и настройка маршрутов.

Для облегчения создания фронтенда сайта производим установку Bootstrap.

Bootstrap (также известен как Twitter Bootstrap) — свободный набор инструментов для создания сайтов и веб-приложений. Включает в себя HTML- и CSS-шаблоны оформления для типографии, веб-форм, кнопок, меток, блоков навигации и прочих компонентов веб-интерфейса, включая JavaScript-расширения.

Так же нужно уточнить необходимые элементы и функционал, который будут реализован на сайте:

- регистрация и аутентификация,
- редактирование данных пользователя,
- пополнение, вывод и отслеживание баланса пользователя,
- создание и просмотр новостей сообщества,
- возможность связи с фондом,
- отслеживание деятельности фонда.

В связи с данными задачами можно составить схему прецедентов для сайта:

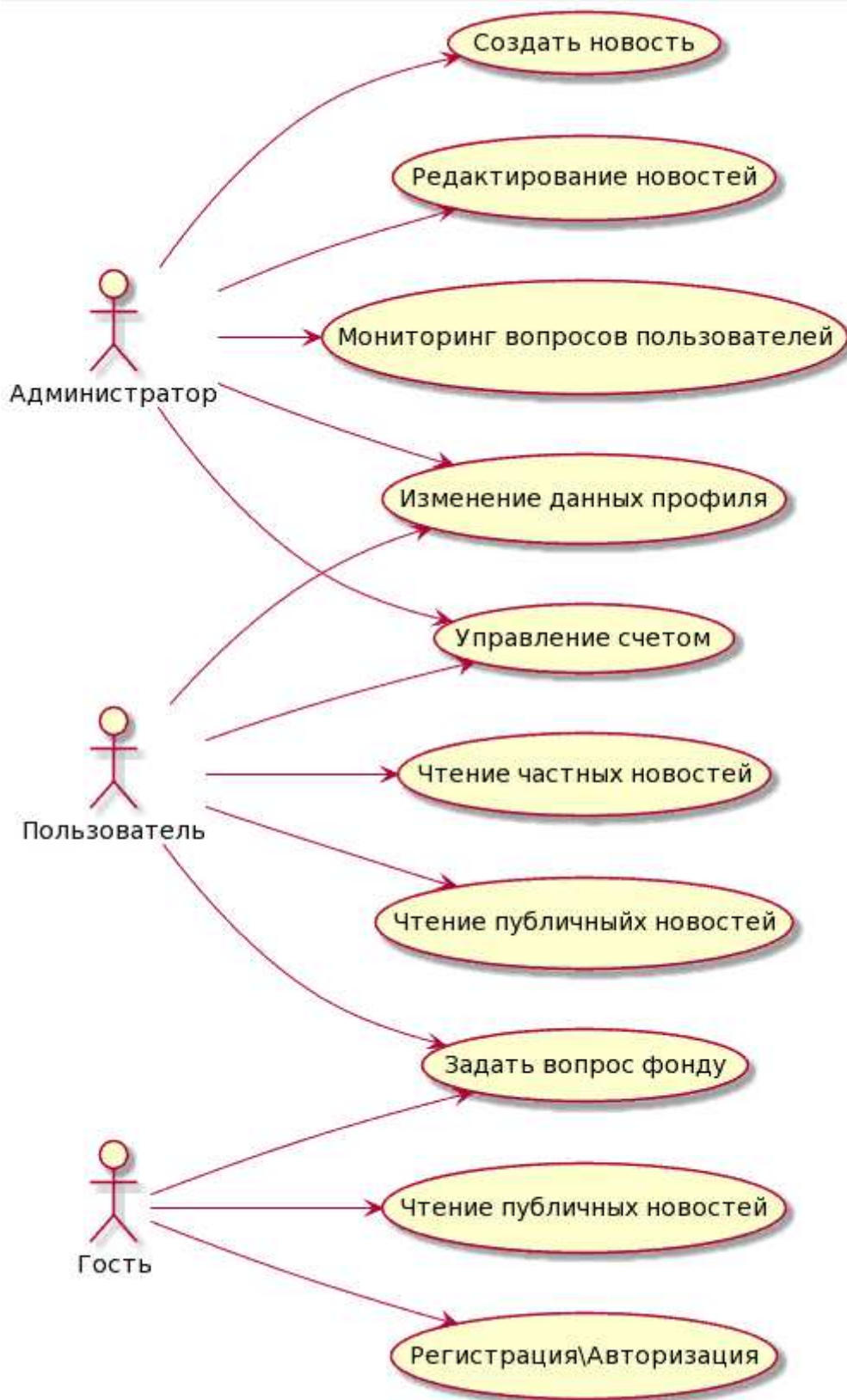


Рисунок 1 – Диаграмма прецедентов

Определившись с основными задачами, необходимо разработать схему сайта. Данная схема будет отображать строение сайта и его логическую часть. Эта схема поможет оценить объем страниц, их логику взаимодействия перед разработкой сайта.

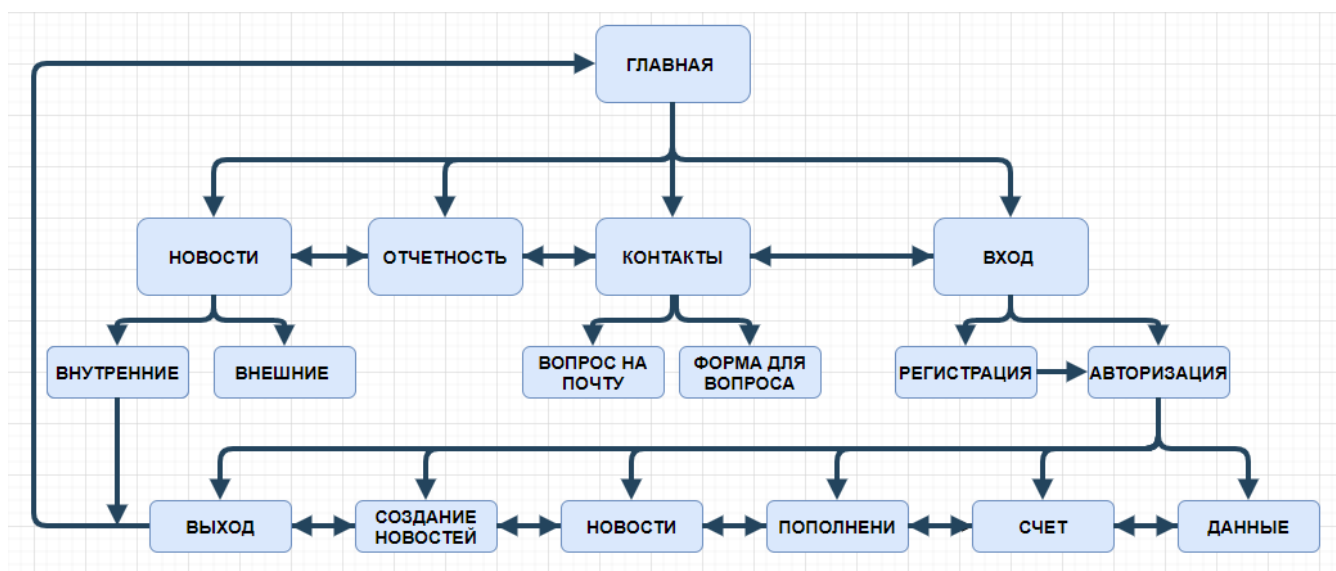


Рисунок 2 - Структура сайта

2.2 Общая схема работы сайта

Фреймворк Laravel написан с использованием шаблона проектирования MVC, то есть «модель-вид-контроллер». Данный шаблон очень популярен и удобен в использовании, так как предусматривает полное отделение логики от его представления. Данное разделение очень удобно, так как каждый элемент шаблона можно править по отдельности, при этом не боясь внести изменения в другие составные части.

Рассмотрим подробнее данный шаблон проектирования.

2.2.1 Модели

В Laravel для работы с базой данных используются так называемые модели (models). Это разновидность ORM (англ. Object-Relational Mapping) и в данном

случае за это отвечает Eloquent ORM. Каждая таблица имеет соответствующий класс-модель, который используется для работы с этой таблицей. Модели позволяют запрашивать данные из таблиц, а также вставлять в них новые записи.

Модель - это правила, используемые для работы с данными, которые представляют концепцию управления сайтом. В модели описываются отношения таблиц.

Создание модели происходит с помощью Artisan-команды, прописываем ключ `-m` для создания соответствующей миграции БД:

```
php artisan make:model 'name' -m
```

В файле миграции необходимо произвести описание таблицы, с которой в дальнейшем будет работать Laravel.

На рисунке приведен пример файла миграции для работы с таблицей «Users» необходимой для регистрации и аутентификации пользователей.

На любом сайте вся структура моделируется как данные, которые обрабатываются определённым образом. С моделями непосредственно работают контроллеры, выполняя запрос из БД или запись, редактирование.

```

2
3 use Illuminate\Database\Migrations\Migration;
4 use Illuminate\Database\Schema\Blueprint;
5 use Illuminate\Support\Facades\Schema;
6
7 class CreateUsersTable extends Migration
8 {
9     /**
10      * Run the migrations.
11      *
12      * @return void
13      */
14     public function up()
15     {
16         Schema::create('users', function (Blueprint $table) {
17             $table->id();
18             $table->string('surname');
19             $table->string('name');
20             $table->string('email')->unique();
21             $table->timestamp('email_verified_at')->nullable();
22             $table->binary('password');
23             $table->rememberToken();
24             $table->timestamps();
25             //$table->string('rules');
26             $table->string('balance')->nullable();
27             $table->string('share')->nullable();
28         });
29     }
30 }
31
32 /**
33  * Reverse the migrations.
34  *
35  * @return void
36  */
37 public function down()
38 {
39     Schema::dropIfExists('users');
40 }
41 }

```

Рисунок 3 - Код миграции для таблицы «Users»

2.2.2 Контроллеры

Вместо того, чтобы определять всю логику обработки запросов в виде функций в файлах роутов, вы можете организовать её с помощью классов контроллеров. Контроллеры могут группировать связанную с обработкой HTTP-запросов логику в отдельный класс.

Создание контроллера происходит с помощью Artisan-команды:

```
php artisan make:controller 'name'
```

Контроллер управляет запросами пользователя (получаемые в виде запросов GET или POST). Его основная функция — вызывать и координировать действие необходимых ресурсов и объектов, и форм нужных для выполнения

действий или запросов, задаваемых пользователем. В контроллере описывается весь необходимый алгоритм действий для определенной модели.

Ниже на рисунке будет представлен код контроллера «PublicNews» который отвечает за чтение формы заполняемой администратором, запись его в БД, а также публикацией новостей в разные разделы сайта, отфильтровывая их по определенному тегу.

```
3 namespace App\Http\Controllers;
4
5 use Illuminate\Http\Request;
6 use App\Http\Requests\PublicNewsRequest;
7 use App\Models\PublicNews;
8 use Illuminate\Support\Facades\Auth;
9 use Models\users;
10
11 class PublicNewsController extends Controller
12 {
13     public function CreateNew(PublicNewsRequest $info) {
14
15         $news = new PublicNews();
16         $news->lent = $info->input('lent');
17         $news->inputimg = $info->input('inputimg');
18         $news->subject = $info->input('subject');
19         $news->message = $info->input('message');
20
21         $news->save();
22
23         return redirect()->back()->with('success', 'Новость опубликованна');
24     }
25
26     public function post_view_new(PublicNewsRequest $info) {
27
28         $news = new PublicNews;
29         return view('news', ['dataN' => $news ->whereIn('lent', ['Общие новости'])
30             -> OrderBy('id','desc')
31             ->take(7)
32             ->get() ]);
33     }
34
35     public function post_priv_new(PublicNewsRequest $info) {
36
37         $news = new PublicNews;
38         return view('account.account', ['dataNP' => $news ->whereIn('lent', ['Частные новости'])
39             -> OrderBy('id','desc')
40             ->take(7)
41             ->get() ]);
42     }
43
44 }
45
```

Рисунок 4 - Код контроллера PublicNews

2.2.3 Маршрутизация

Маршрутизация (Routers) несет на себе основную задачу. С помощью маршрутизации мы описываем, какому действию пользователя соотнести какой

контроллер, форму или просто перенаправить его на сопутствующую страницу или раздел сайта.

Все роуты Laravel определены в файлах роутов, которые расположены в директории routes. Эти файлы автоматически загружаются фреймворком. В файле routes/web.php определены роуты для нашего web-интерфейса. Эти роуты входят в группу посредников web, которые обеспечивают такие возможности, как состояние сессии и CSRF-защита.

```
3 use Illuminate\Support\Facades\DB;
4 use Illuminate\Support\Facades\Route;
5 use Illuminate\Support\Facades\Auth;
6
7 Route::get('/', function () {
8     return view('welcome');
9 }) ->name ('welcome');
10
11
12 Route::get('/Contacts', function () {
13     return view('Contacts');
14 }) ->name ('contacts');
15
16 /* Регистрация */
17 Route::get('/Registr', function () {
18     return view('Registr');
19 }) ->name ('registr');
20
21
22 Route::get('/Account', 'AccountController@Getmi')
23 ->name ('account') ;
24
25 // Вывод вопросов пользователей
26 Route::get('/Account.inc.questions', 'QuestionController@OutQuestions')
27 ->name ('oquestion') ;
28
29
30 /*Вывод отчета о торговле*/
31 Route::get('/Prognoz', 'OtchetController@post_otchet')
32 ->name ('otchet') ;
```

Рисунок 5 - Часть кода из файла web.php .

2.2.4 Blade шаблоны

Blade — простой, но мощный шаблонизатор, поставляемый с Laravel. В отличие от других популярных шаблонизаторов для PHP Blade не ограничивает нас в использовании чистого PHP-кода в ваших представлениях. На самом деле все представления Blade скомпилированы в чистый PHP-код и кешированы, пока в них нет изменений, а значит, Blade практически не нагружает наш сайт. Файлы

представлений Blade используют расширение `.blade.php` и обычно хранятся в папке `resources/views`.

Таким образом можно создавать отдельные шаблоны страниц, элементов интерфейса. При необходимости подключать шаблоны к интересующим нас страницам.

На рисунке показан один из макетов основной разметки страницы, к которому подключаются необходимые компоненты.

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <meta charset="UTF-8">
5 <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
6 <title>@yield('title')</title>
7 <link rel="stylesheet" href="css/app.css">
8 <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/css/bootstrap.min.css">
9 <link rel="shortcut icon" href="{{ asset('img/logo/logo.png') }}">
10 </head>
11
12
13 <body>
14     @if (Auth::check())
15         @include('account.header')
16
17     @else
18         @include('inc.header')
19
20     @endif
21
22     @include('inc.messagesSuc')
23
24     <div class="container ">
25
26         <div class="row">
27             <div class="col-8 shadow p-3 mb-5 rounded-lg ">
28                 @yield('content')
29             </div>
30
31             <div class="col-3 shadow p-3 ml-4 mb-5 rounded-lg" style="height: 500px;">
32                 @include('inc.aside')
33             </div>
34
35         </div>
36
37     </div>
38     @include('inc.footer')
```

Рисунок 6 - Подключение контента к основному макету

Веб-приложение обычно состоит из набора контроллеров, моделей и форм. Проект разделен на несколько важных частей – модели, контроллеры, маршруты и формы, а за внешнее составляющее отвечают blade шаблоны.

форму или просто перенаправить его на сопутствующую страницу или раздел сайта.

С помощью схемы можно отобразить систему работы фреймворка Laravel для реализации сайта. Где будет показана зависимость основных элементов.

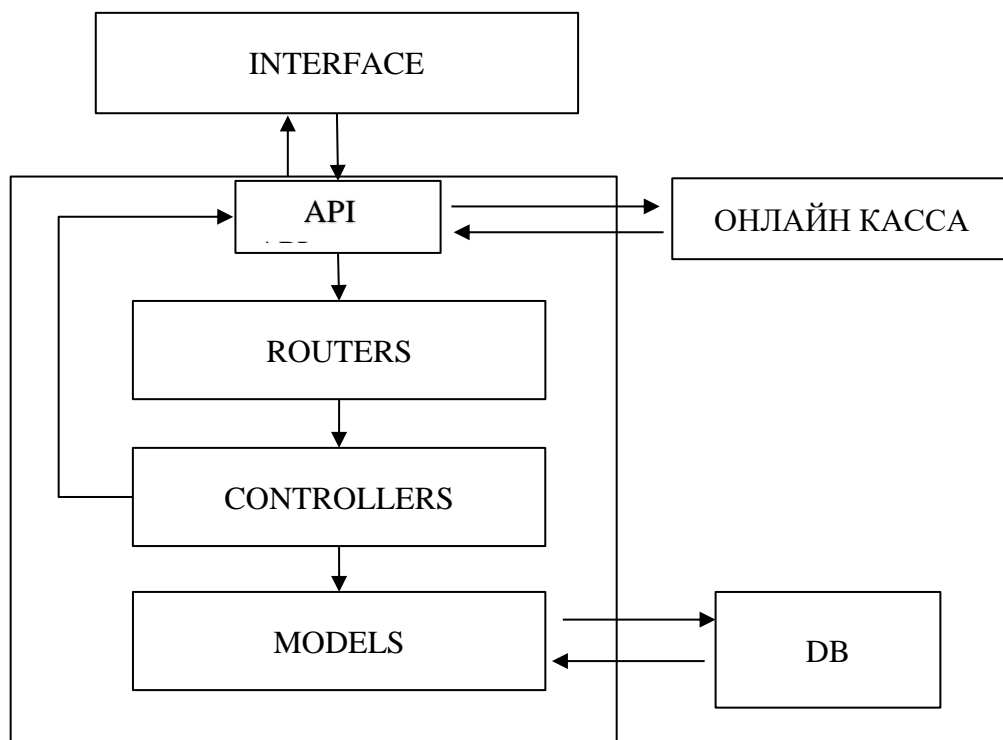


Рисунок 7 - Системная схема работы и управления фреймворка Laravel

Для реализации данного сайта, будет использоваться БД с большим количеством таблиц. В таблицах будут храниться данные пользователей, состояния их счета, вопросы в службу поддержки, новостная лента и много сопутствующей информации. Необходимо создать схему БД, в которой будут показаны необходимые таблицы, а также их зависимости.

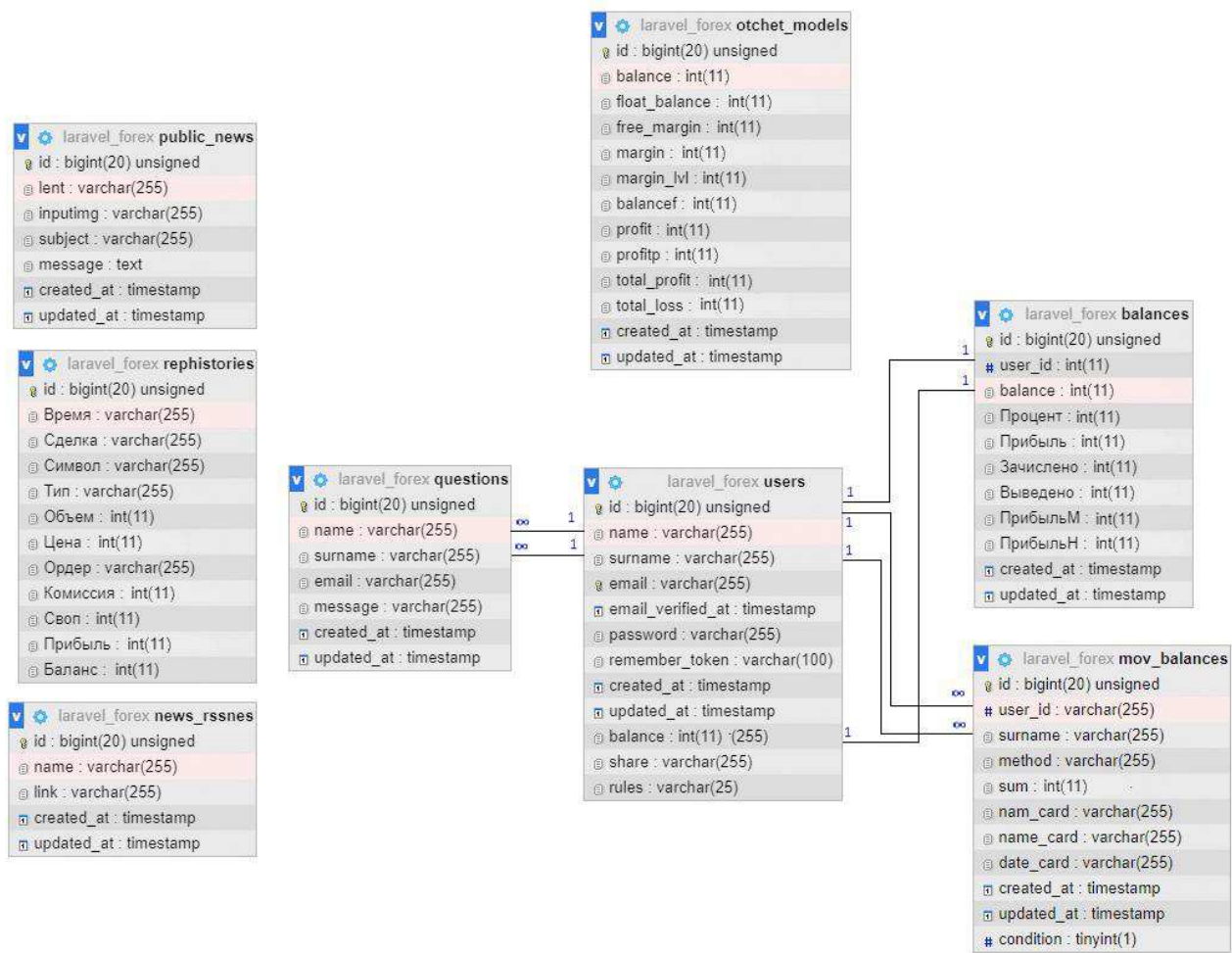


Рисунок 8 – Схема взаимосвязей БД

3. Разработка сайта и описание сайта

3.1 Главная страница

Так как мы разрабатываем сайт для паевого фонда, в первую очередь необходимо познакомить пользователей с ПИФ. Рассказать про основное направление, показать сильные стороны компании, заинтересовать и показать работу маркетологов.

По такому случаю первая и заглавная страница является визиткой, на которой изложена первичная информация.

Один из пунктов нашего задания является простота и лаконичность, сайт не загромождён лишней информацией. Так как главная страница (визитка) является всего лишь ознакомительной. На ней не присутствует боковое меню, для более простого восприятия информации.

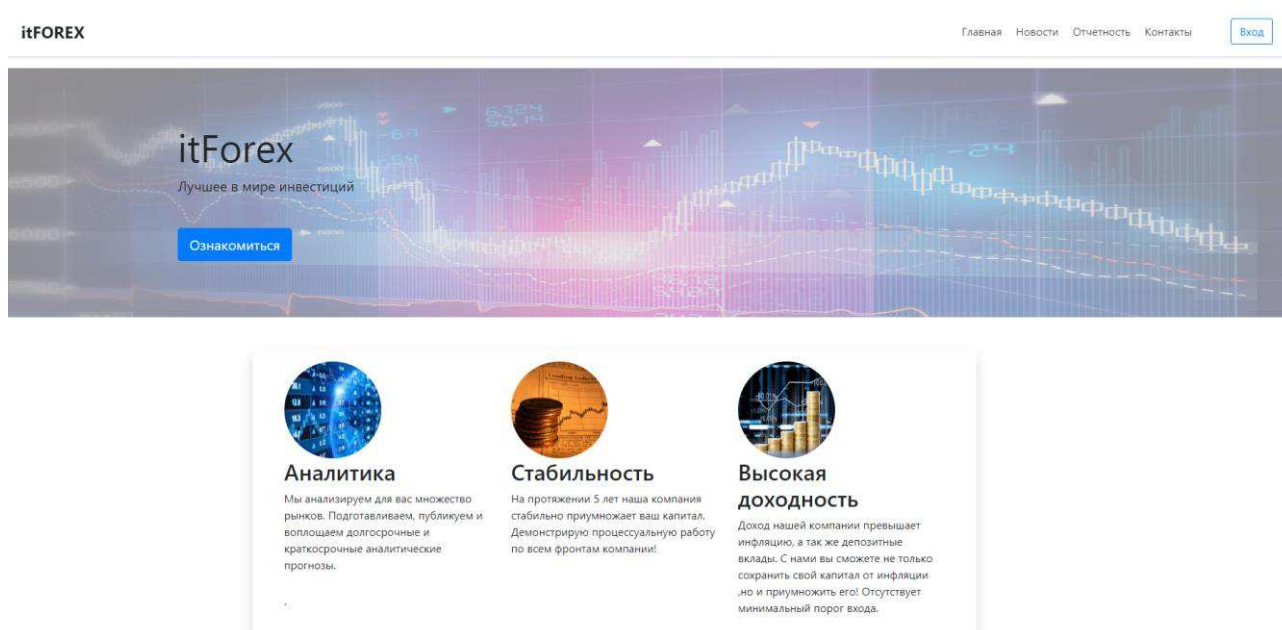


Рисунок 9 — Главная страница

3.2 Header-основное меню сайта

В верхней части сайта располагается наше меню. В котором мы можем увидеть все необходимые вкладки: «Главная страница», «Новости», «Отчет», «Контакты» и «Вход» в личный кабинет.

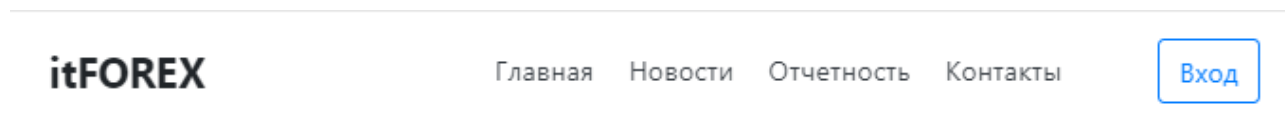


Рисунок 10 – Навигационное меню сайта

Для нашего сайта потребуется огромное количество страниц, чтобы не писать в каждом файле множество однотипного текста (на примере Header или стандартной HTML разметки и конструкции страницы). Я создаю blade-шаблоны страниц и отдельных элементов интерфейса, в дальнейшем подключаю эти элементы и собираю в единое целое.

3.3 Footer – нижний колонтитул сайта.

Нижний колонтитул сайта, как и с основным меню, подключаемый.

На нем мы можем увидеть второстепенную информацию, а также значок письма, что позволяет моментально отправить письмо с любым интересующим вас вопросом. Иконки социальных сетей, где также может располагаться полезная информация о сообществе, а также свежие новости. В крайнем правом углу есть маркер «Возврат к началу».

3.4 Новости

На данной вкладке представлены основные новости фонда, аналитика, а также основные направления развития фонда.

Данная страница полностью шаблонна. Администраторы сайта из личного кабинет могут с легкостью добавлять новости на данную страницу, а также редактировать их.

Так же на этой странице мы уже можем заметить боковую панель, в котором отображаются важные новости мировой биржи. В данный момент идет трансляция RSS новостей с «Московской биржи». После аутентификации в личном кабинете можно будет выбирать определенные источники новостных вещаний.

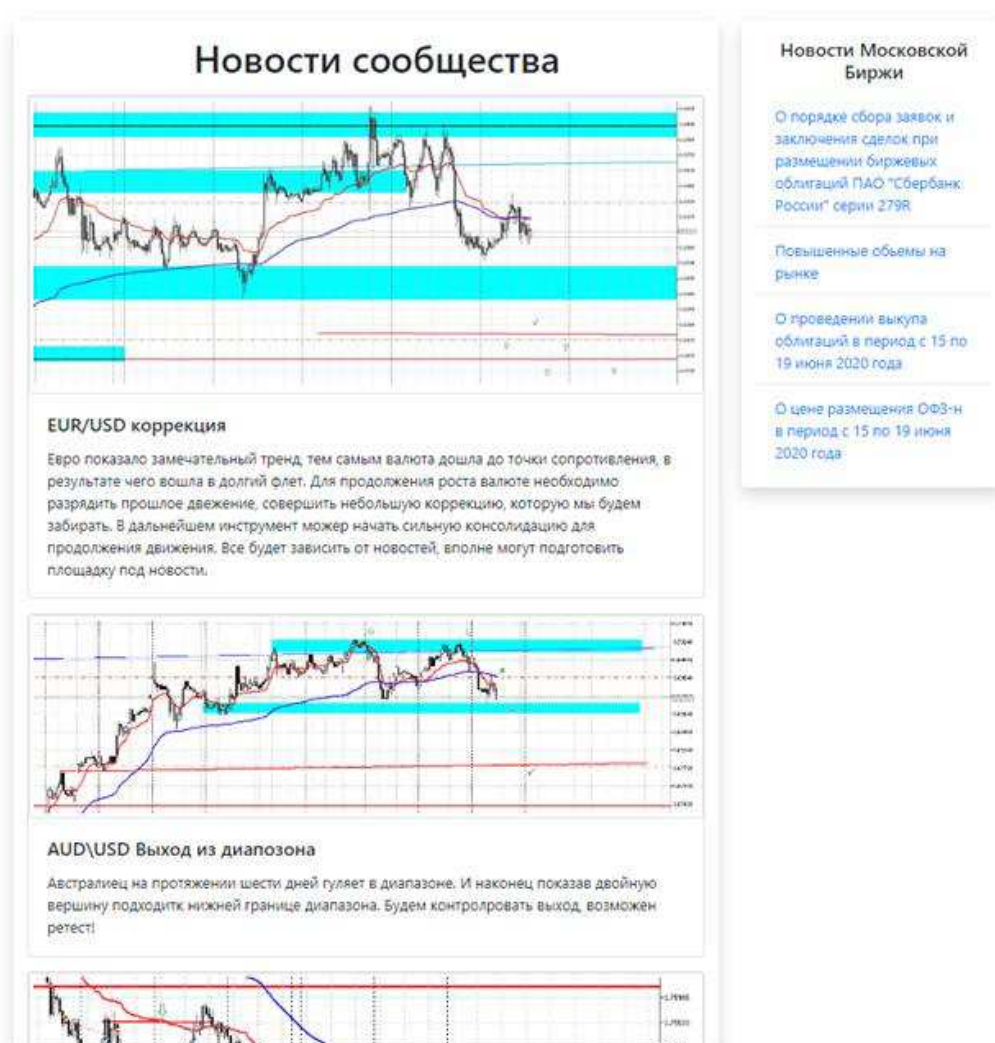


Рисунок 11 – Вкладка «Новости»

3.5 Отчет фонда

На данной странице представлен полный отчет о проделанной работе ПИФ. Подробный график изменения баланса, отчетность баланса, прибыли и убытки и т.д. Все данные интегрированы из обновляемых БД, что добавляет сайту автономности. В БД непосредственно происходит импорт определенных таблиц.

Обновление отчетности по некоторым соображениям происходят пару раз в сутки, в зависимости от активности ПИФ. Опять же эта информация не обновляется ежеминутно из определенных соображений по сохранению конфиденциальности и ресурсов сайта.

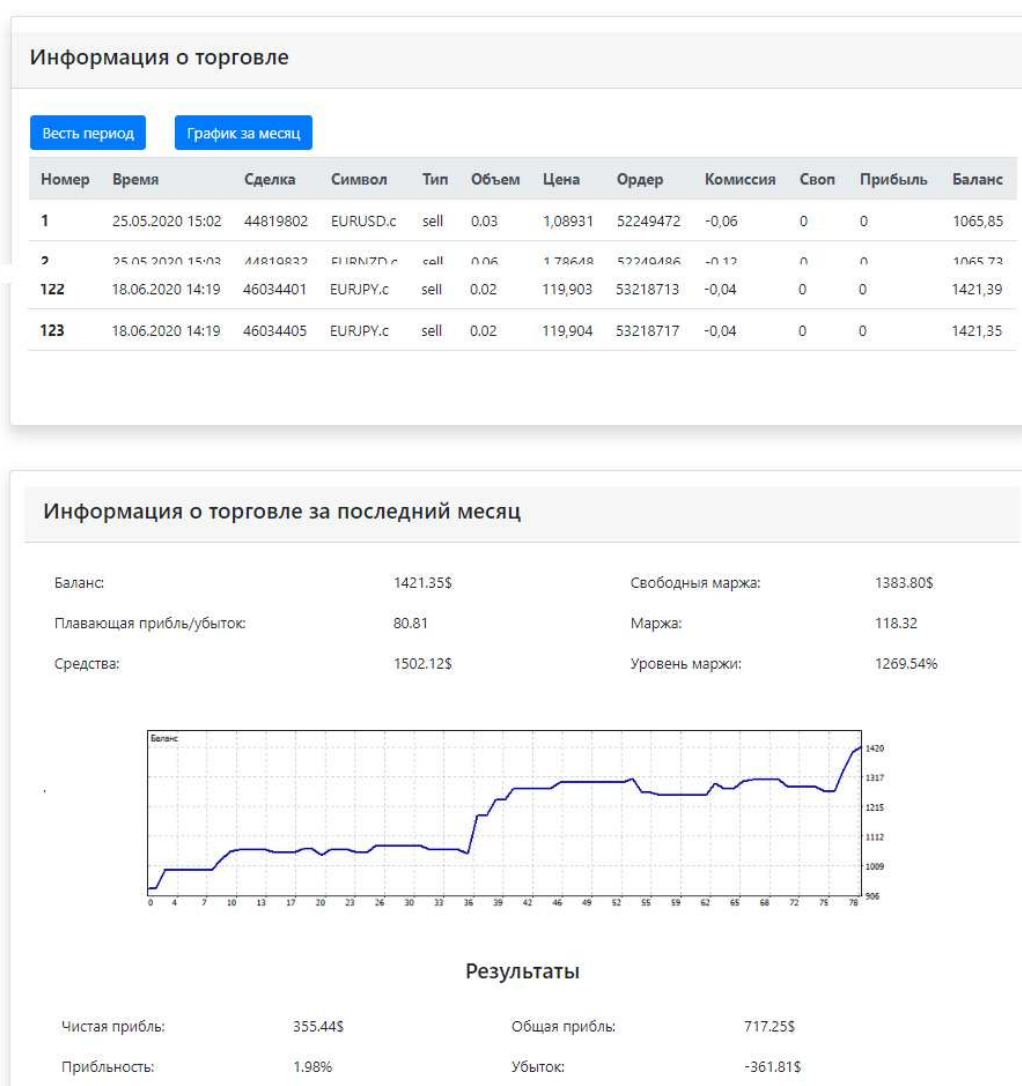


Рисунок 12 – Вкладка «Отчет»

Но зарегистрированные пользователи на данной вкладке увидят специальную интегрированную ссылку и подробную инструкцию, о подключении в терминале или в мобильном приложении к «инвестиционному» аккаунту ПИФ. Где они смогут отслеживать в реальном времени все возможные рынки ПИФ, а также все выполняемые или отложенные ордера ПИФ. Что будет реализовывать не только ежедневное отслеживание отчетности при просмотре сайта, а также постоянный мониторинг деятельности ПИФ.

3.6 Контакты

В этом разделе можно наблюдать не только важную информацию для связи с ПИФ, такую как телефон или ссылки для отправки писем в разные разделы фирмы. Но и раздел для написания срочного вопроса в службе поддержки, также это первая форма для ввода информации, которую обрабатывает контроллер и заполняет БД. В дальнейшем эти сообщения смогут просмотреть администраторы.

Контакты

Мы всегда будем рады ответить вам на любые вопросы связанные с нашей компанией, вкладами, аналитикой, и перспективами роста. А так же на сопутствующие вопросы.

Телефон

Телефон горячей линии: +7-(913)-521-69-38

Почта

Задайте вопрос аналитикам	" Отдел аналитики "
Задайте вопрос менеджерам	" Отдел управления капиталом "
Задайте интересующие вас вопросы	" Отдел работы с клиентами "
Задайте вопрос в службу поддержки	" Служба поддержки "

Фамилия

Имя

Адрес почты

Ваше сообщение

Рисунок 13 – Вкладка «Контакты»

3.7 Регистрация и аутентификация

В данный момент регистрация представлена отдельной формой.

• Обязательно впишите свою фамилию
• Обязательно впишите свое имя
• Обязательно впишите свою почту
• Обязательно впишите пароль

Регистрация

Фамилия

Имя

Адрес почты

Пароль

Проверка пароля

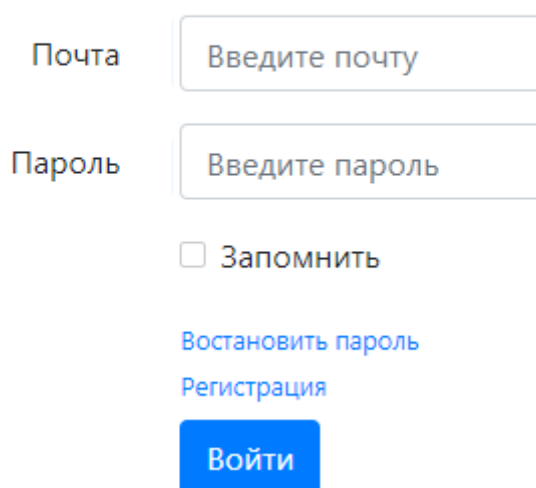
[Вход в личный кабинет](#)

Рисунок 14 - Форма регистрации

Данная форма имеет все основные, первичные данные необходимые для создания личного кабинета инвестора: «Фамилию», «Имя», «Почту», «Пароль», «Проверку пароля».

После заполнения всех ячеек формы, контролер обращается к функции «Request» в которой описаны определённые правила для каждой ячейка. На фото представлены часть ошибок. Также форма проверяет на соответствие правильному написанию адреса почты, на соответствие длинны и сравнения паролей. Если контроллер заметит, что данная почта уже присутствует в БД, она предложит пройти аутентификацию. Данная форма отправляет данные в БД, создавая там новую запись о пользователе. После успешного завершения, пользователя будет перенаправлен на страницу аутентификации с сообщение об успешном завершении регистрации.

Авторизация



Почта

Пароль

Запомнить

[Восстановить пароль](#)

[Регистрация](#)

Рисунок 15 – Форма авторизации на сайте

Данная форма также имеет подключенную функцию проверки заполнения, эта форма запрашивает и сравнивает данные из БД. При успешном совпадении почты и пароля, аутентификация считается успешно пройденной. Система переводит пользователя в личный кабинет. В дальнейшем сайт будет работать с данными пользователя.

3.8 Личный кабинет

После авторизации происходит изменение навигационного меню. Вместо кнопки «Вход» появилась фамилия авторизованного пользователя, а также расширенное меню, которое позволяет быстро, из любого раздела сайта перенести нас на нужную вкладку личного кабинета.

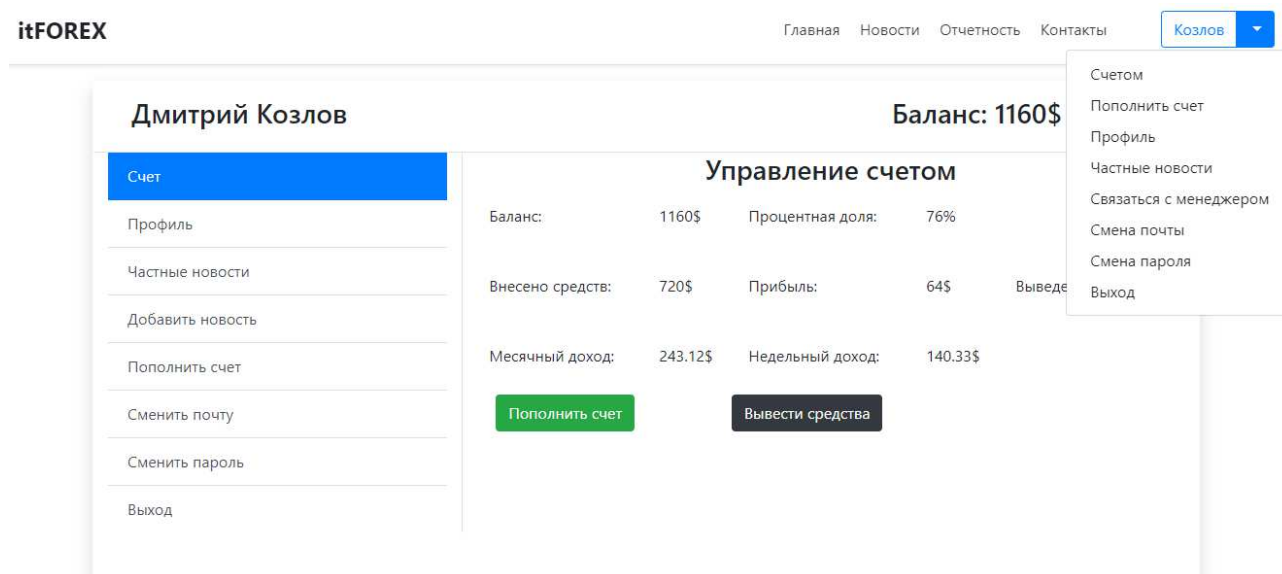


Рисунок 16 – Личный кабинет инвестора

Личный кабинет выполнен в информативной форме. На основной странице в верхней части можем наблюдать инициалы владельца аккаунта и его баланс. Так же первой вкладкой в меню личного кабинета открыта вкладка с информацией о балансе. Все данные заполняются из БД определенного пользователя.

Работоспособность навигационного меню в личном кабинете пользователя реализовано с помощью JavaScript, таким образом пользователь может переключаться по разделам меню, не ожидая и не прерывая рабочий процесс на перезагрузку страницы.

В меню пользователя представлены все необходимые вкладки для управления своим аккаунтом. В «Редакторе личных данных» можно произвести

изменения и дополнении информации о себе. Все ранее заполненные данные будут выгружаться из БД и прописываться в необходимых строках.

Имеется раздел частных новостей, это более подробный отчет о перспективах и аналитических решениях компании.

Стоит упомянуть о разделе «Добавить новость». Данный раздел доступен только администраторам сайта, реализуется это за счет дополнительной проверки данных пользователя и предоставления ему дополнительных пунктов меню.

Дмитрий Козлов Баланс: 1160\$

Счет

- Профиль
- Частные новости
- Добавить новость
- Пополнить счет
- Сменить почту
- Сменить пароль
- Выход

Профиль

Имя: Фамилия: Отчество:

Почта: Телефон:

Паспорт

Код подразделения: Серия/номер:

Страна: Край:

Город: Адрес:

Индекс:

Рисунок 17 – Форма заполнения данных о инвесторе

В этом разделе Администратор может добавить новость в необходимый раздел новостной ленты: Общая или Частная. Прикрепить изображение, написать тему и заполнить основную часть повествования. После чего все данные заносятся в БД. После чего другой контроллер выводит эти данные в форму и на сайте опубликуется новая новость.

Рисунок 18 – Форма создание администратором новых новостей

Присутствует вкладка пополнения баланса. Смена почты и пароля перед принятием данных требует подтверждения с почты для проверки запроса (безопасности).

Рисунок 19 – Форма для пополнения счета

4. Заключение

В результате анализа задания работы и систем CMS было принято решение написать сайт при помощи фреймворка Laravel и всех необходимых комплектующих для реализации его функционала.

Была достигнута требуемая цель и выполнены поставленные задачи:

- разработка шаблона с доступным неперегруженным интерфейсом;
- возможность следить за новостями фонда и фондовых рынков;
- реализовать возможность отслеживания успехов фонда;
- разработать удобный и информативный личный кабинет для пользователя;

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Википедия -Свободная энциклопедия : URL - <https://ru.wikipedia.org> ;
2. 1С-Битрикс Система создания и управления сайтом: URL - <https://www.1c-bitrix.ru/>;
3. WordPress Система создания и управления сайтом : URL - <https://ru.wordpress.org> ;
4. Joomla Система создания и управления сайтом : URL - <https://joomla.ru> ;
5. Drupal Система создания и управления сайтом : URL - <https://drupal.ru> ;
6. Laravel Фреймворк : URL - <https://laravel.com> ;
7. Laravel RU Форум и документация : URL - <https://laravel.ru> ;
8. BootStrap шаблоны для HTML и CSS : URL - <https://getbootstrap.com> ;
9. BootStrap RU документация: URL - <https://bootstrap-4.ru> .

Приложение А

Маршруты:

```
/* Создание Новостей*/
Route::post('/Account.add_new', 'PublicNewsController@CreateNew')
->name ('create-new');

/*Публикация новостей общих*/
Route::get('/news', 'PublicNewsController@post_view_new')
->name ('news');

/*Публикация новостей частных*/
Route::get('/Account.priv_news', 'PublicNewsController@post_priv_new')
->name ('priv_news');
```

Файл миграции:

```
<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

class CreatePublicNewsTable extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('public_news', function (Blueprint $table) {
            $table->id();
            $table->string('lent');
            $table->string('inputing');
            $table->string('subject');
            $table->string('message');
            $table->timestamps();
        });
    }
    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::dropIfExists('public_news');
```

```
}  
}
```

Контроллер:

```
namespace App\Http\Controllers;  
  
use Illuminate\Http\Request;  
use App\Http\Requests\PublicNewsRequest;  
use App\Models\PublicNews;  
use Illuminate\Support\Facades\Auth;  
use Models\users;  
  
class PublicNewsController extends Controller  
{  
    public function CreateNew(PublicNewsRequest $info) {  
  
        $news = new PublicNews();  
        $news->lent = $info->input('lent');  
        $news->inputimg = $info->input('inputimg');  
        $news->subject = $info->input('subject');  
        $news->message = $info->input('message');  
  
        $news->save();  
  
        return redirect()->back()->with('success', 'Новость опубликована');  
    }  
  
    public function post_view_new(PublicNewsRequest $info) {  
  
        $news = new PublicNews;  
        return view('news', ['dataN' => $news ->whereIn('lent', ['Общие новости'])  
            -> OrderBy('id', 'desc')  
            ->take(7)  
            ->get() ]);  
  
    }  
  
    public function post_priv_new(PublicNewsRequest $info) {  
  
        $news = new PublicNews;  
        return view('account.account', ['dataNP' => $news -  
>whereIn('lent', ['Частные новости'])  
            -> OrderBy('id', 'desc')  
            ->take(7)  
            ->get() ]);  
    }  
  
}
```

Код вывода новостей:

```
@foreach( $dataNP as $el)

    <div class="card my-3" ">
        
        <div class="card-body">
            <h5 class="card-title">{{$el->subject}}</h5>
            <p class="card-text">{{$el->message}}</p>
            <small class="text-muted">Дата публикации: {{$el-
>created_at}}</small>
        </div>
    </div>

@endforeach
```

Форма создания новостей:

```
<form action="{{route('create-new')}}" method="post" >
    @csrf

    <div class="form-group w-25">
        <label for="lent"> Выбор новостной Ленты</label>
        <select class="form-
control" name="lent" id="lent" placeholder="Выбор новостной Ленты">
            <option>Общие новости</option>
            <option>Частные новости</option>
        </select>
    </div>

    <p>Загрузка изображения в шапку новости</p>

    <div class="input-group w-50 mb-3">
        <div class="custom-file">

            <input type="file" class="custom-file-
input" name="inputimg" id="inputimg" type="file" multiple name="file[]" placeholder
="Загрузка изображения" aria-describedby="inputGroupFileAddon01">

        </div>
    </div>

    <input type="file" name="inputimg" multiple name="file[]" class="form-
control-file" id="inputimg">
```

```
<div class="form-group w-75">
  <label for="subject">Тема новости</label>
  <input type="text" class="form-control" name="subject" id="subject" aria-
describedby="">

</div>
<div class="form-group w-75 ">
  <label for="message">Новость</label>
  <textarea class="form-
control" name="message" id="message" rows="5"></textarea>
</div>
<button type="submit" class="btn btn-success">Опубликовать</button>

</form>
```

Приложение В

<ul style="list-style-type: none">Controllers<ul style="list-style-type: none">Auth<ul style="list-style-type: none">ConfirmPasswordController.phpForgotPasswordController.phpLoginController.phpRegisterController.phpResetPasswordController.phpVerificationController.phpAccountController.phpController.phpHomeController.phpLoginController.phpLogoutController.phpNewsRssController.phpOtchetController.phpPublicNewsController.phpQuestionController.phpRephistoryController.phptest-HomeController.phpUsersRegistrationController.phpMiddleware<ul style="list-style-type: none">Authenticate.phpCheckForMaintenanceMode.phpEncryptCookies.phpRedirectIfAuthenticated.phpTrimStrings.phpTrustHosts.phpTrustProxies.phpVerifyCsrfToken.phpRequests<ul style="list-style-type: none">LoginRequest.phpPublicNewsRequest.phpQuestionsRequest.phpUsersRegistrationRequest.phpKernel.phpModels<ul style="list-style-type: none">balance.phpContact.phpMov_balance.phpNewsRss.phpOtchetModel.phpPublicNews.phpQuestion.phpRephistory.phpUsers.phpProviders<ul style="list-style-type: none">AppServiceProvider.phpAuthServiceProvider.phpBroadcastServiceProvider.phpEventServiceProvider.phpRouteServiceProvider.phpUser.php	<ul style="list-style-type: none">Models<ul style="list-style-type: none">balance.phpContact.phpMov_balance.phpNewsRss.phpOtchetModel.phpPublicNews.phpQuestion.phpRephistory.phpUsers.phpProviders<ul style="list-style-type: none">AppServiceProvider.phpAuthServiceProvider.phpBroadcastServiceProvider.phpEventServiceProvider.phpRouteServiceProvider.phpUser.phpbootstrapconfigdatabase<ul style="list-style-type: none">factoriesmigrations<ul style="list-style-type: none">2014_10_12_000000_create_users_table.php2014_10_12_100000_create_password_resets_table.php2019_08_19_000000_create_failed_jobs_table.php2020_06_09_073022_create_contacts_table.php2020_06_09_184827_create_news_rsses_table.php2020_06_15_134323_create_public_news_table.php2020_06_15_155637_create_questions_table.php2020_06_17_203810_create_otchet_models_table.php2020_06_19_070014_create_rephistories_table.php2020_06_21_194039_create_balances_table.php2020_06_22_103800_create_mov_balances_table.php	<ul style="list-style-type: none">views<ul style="list-style-type: none">account<ul style="list-style-type: none">inc<ul style="list-style-type: none">add_new.blade.phpbalanc.blade.phpchange_email.blade.phpchange_password.blade.phppriv_news.blade.phpprofile.blade.phpquestions.blade.phpup_balance.blade.phpAccount.blade.phpheader.blade.phpIndex.blade.phpauth<ul style="list-style-type: none">passwordslogin.blade.phpregister.blade.phpverify.blade.phphistoryinc<ul style="list-style-type: none">aside.blade.phpfooter.blade.phpforma.blade.phpheader.blade.phpmessagesErr.blade.phpmessagesSuc.blade.phplayouts<ul style="list-style-type: none">app.blade.phpCor-aside.blade.phpCor-noside.blade.phpContacts.blade.phpLogin.blade.phpNews.blade.phpPrognoz.blade.phpRegistr.blade.phpWelcome.blade.phproutes<ul style="list-style-type: none">api.phpchannels.phpconsole.phpweb.php
---	---	--

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий
институт

Вычислительная техника
кафедра

УТВЕРЖДАЮ

Заведующий кафедрой


О. В. Непомнящий

подпись

инициалы, фамилия

« _____ » _____ 2020г.

БАКАЛАВРСКАЯ РАБОТА

09.03.01 – «Информатика и вычислительная техника»

код - наименование направления

Веб-сайт для инвестирования и мониторинга фондового рынка

тема

Руководитель



подпись, дата

канд. техн. наук, доцент

должность, ученая степень

М. С. Медведев

инициалы, фамилия

Выпускник



подпись, дата

Д. И. Козлов

инициалы, фамилия

Нормоконтролер



подпись, дата

канд. техн. наук, доцент

должность, ученая степень

М. С. Медведев

Красноярск 2020