**Содержание**

[Введение 5](#_Toc40794454)

[1. Модификация информационной системы 6](#_Toc40794455)

[1.1 Описание предметной области 6](#_Toc40794456)

[1.2 Описание входных и выходных данных 12](#_Toc40794457)

[1.3 Проектирование диаграммы прецедентов и схемы данных 14](#_Toc40794458)

[1.4 Выбор программных и технических средств разработки 19](#_Toc40794459)

[2. Разработка программных решений 21](#_Toc40794460)

[3. Тестирование программного продукта. 24](#_Toc40794461)

[Заключение 25](#_Toc40794462)

[Список использованных источников 26](#_Toc40794463)

# **Введение**

Темой преддипломной практики является модификация системы по учету и анализу рабочего времени на предприятии. Актуальность данной темы состоит в том, что неэффективное использование рабочего времени наносит большой ущерб в экономической сфере предприятий. Живой труд, в отличие от вещественного, невозможно накапливать, для использования тогда, когда появится в нем потребность. Нерациональное использование рабочего времени сотрудников в рыночных условиях может свести к минимуму преимущества, которые дают современные технологии и различная техника. Помимо этого, актуальность проекта в этой области можно определить не только напряженностью трудовых ресурсов, но и тем, что упадок производительности труда объясняется нерациональным их использованием. Устранение и сведение к минимуму недостатков в использовании рабочего времени - важное условие обеспечения роста объемов производства, в связи с котором отпадает необходимость увеличения сотрудников предприятий. В связи с этим возрастает значение изучения уровня эффективности использования рабочего времени, динамики и структуры его потерь на предприятии, научной и программной разработки путей их устранения.

Целью исследования является разработка информационной системы по учету и анализу рабочего времени на предприятии.

В задачи по разработки дипломного проекта входят:

* Анализ предметной области и постановка задачи
* Описание входных и выходных данных
* Проектирование диаграммы прецедентов и схемы данных
* Выбор программных и технических средств разработки
* Разработка программных решений
* Разработка структуры базы данных
* Основные этапы программирования приложения
* Разработка интерфейса пользователя
* Обеспечение информационной безопасности разработанной информационной системы
* Тестирование и документирование программных решений
* Разработка тест-кейсов
* Требования к технической документации при проектировании и эксплуатации системы
* Экономическая эффективность внедрения программных решений
* Разработка мобильного приложения
* Создание руководства пользователя
* Создание руководства программиста

Объектом исследования является рабочее время на предприятиях и организациях нефтяной промышленности.

Предметом исследования является информационная система по учету и анализу рабочего времени на предприятии.

# **1. Модификация информационной системы**

# **1.1 Описание предметной области**

Информационная система (ИС) — система, предназначенная для хранения, поиска и обработки информации, и соответствующие организационные ресурсы (человеческие, технические, финансовые и т. д.), которые обеспечивают и распространяют информацию (ISO/IEC 2382:2015). ИС предназначена для своевременного обеспечения надлежащих людей надлежащей информацией, то есть для удовлетворения конкретных информационных потребностей в рамках определённой предметной области, при этом результатом функционирования информационных систем является информационная продукция — документы, информационные массивы, базы данных и информационные услуги.

Рабочее время является основной экономической категорией, всеобщим измерителем затрат труда на осуществление разнообразных производственных процессов. На предприятиях все трудовые, технологические и производственные процессы функционируют в пространстве и времени. Именно поэтому рабочее время служит не только главным производственным ресурсом, но и критерием экономической эффективности производства. В рыночных экономических отношениях рабочее время является также универсальным критерием рациональности всех действующих и проектируемых производственных и организационных систем. Любая система организации труда и производства признается наиболее экономичной, если ее существование при прочих равных условиях происходит с минимальными затратами производственных ресурсов и рабочего времени. При этом цена рабочего времени в каждой системе определяется в основном стоимостью так называемых фондообразующих факторов, состав которых ограничивается содержанием производственных процессов. Цена времени в целом соответствует совокупному потенциалу развития каждой производственной или организационной системы.

Предприятия нефтяной промышленности - занимаются не только лишь добычей нефти, но еще и ее транспортировкой. Помимо этого, осуществляется и добыча попутного газа. На территории Российской Федерации имеется довольно много уже найденных запасов нефти. Это позволяет России занимать шестое место в мире в данной отрасли. Больше всего изучены и освоены природные ресурсы в Волго-Уральском регионе. Именно тут находятся самые крупные месторождения нефти. Но основные ресурсы сосредотачиваются в Западно-Сибирском округе. Не останавливаются формироваться и Тимано-Печорские нефтедобывающие базы. Тут происходит добыча так называемой «тяжелой» нефти при помощи шахтного способа. Это очень ценное сырье, из которого получается низкотемпературное масло, которое крайне необходимо для работы оборудования в суровом климате. Приватизация объектов нефтегазового комплекса (НГК) разделила до этого единую систему, управляемую государством. Частные нефтяные предприятия завладели всеми производственными объектами и национальным богатством России – месторождениями и запасами нефти. В НГК присутствует 17 компаний. Из-за усиленной добычи нефтепродуктов в восточных регионах, а также на севере, возникла проблема их транспортировки. Самым эффективным решением стали трубопроводы. А вот благодаря развитию сети нефтепроводов стало возможно приближение нефтяной переработки к местам ее потребления. Размещение предприятий НГК напрямую связано с объемами потребления нефтепродуктов в различных областях, самой техники переработки, перевозки нефти, а также территориальных соотношений между местами потребления и ресурсами.

Информационная система отвечает всем требованиям организаций, таким как быстродействие, плавность интерфейса, стабильность работы, поддержка многопользовательского режима, многофункциональность, приложение с возможностью дальнейшей модификации.

Реализация системы возможна с использовнием технологий и не выходят за пределы заданной стоимости т.к технологии бесплатны и на установку уходит минимальное количество времени, а оплата происходит только за базовые затраты.

Присутствует возможность объединить ИС с уже эксплуатируемыми системами, т.к. использующийся в системе фреймворк Vue.js можно интегрировать в любой возможный веб - проект как обычный язык разметки HTML а Php в любой модуль для подержания веб – части.

Если ИС не будет введена в эксплуатацию в организации или предприятии, тогда, возможно, неэффективное использование рабочего времени нанесет большой ущерб в экономической сфере и приведет к тому, что предприятие будет затрачивать больше экономических ресурсов, для которого такое решение является нецелесообразным.

Речь о создании ИС зародилась после того, как была раскрыта актуальность данной темы для эффективного внедрения всех ресурсов, увеличения значения управленческих принимаемых решений, ориентированных на повышение доходов.

Также ИС реализовывает различные бизнес-процессы, такие как:

* разгрузка обязанностей для руководителей;
* экономия финансовых средств;
* экономия времени с внедренной системой;
* назначение задач использующим систему;
* мониторинг процессов;
* организационная структура;

Порой, в предприятиях и организации возникают различные проблемы, а внедрение ИС позволяет их решить.

В основном это распространенные проблемы, такие как:

* проблема доступности данных
* проблема полноты информации
* проблема актуализации информации
* проблема единых стандартов и технологий
* проблемы информационной безопасности
* проблема исторических данных

Проблема доступности данных, равно как и оперативности доступа к этим данным практически отсутствует в случае, если сотрудники вашей компании работают с единой информационной системой. В любой момент времени, любой сотрудник может создать, изменить или просмотреть требуемую информацию. Для формирования требуемого отчета необходимы считаные секунды - вся информация хранится в базе данных и доступна в любое время.

Еще одна проблема, с которой сталкиваются руководители, да и рядовые сотрудники тоже - это полнота данных. Проблема выражается в том, что для принятия управленческого недостаточно данных (например, не ведется учет по источникам информации, откуда клиенты узнали о компании), данные не достаточно качественные (одни менеджеры интересуются персональной информацией контактных лиц, другие - нет), данные разрозненные.

Применение информационной системы позволяет решить эту проблему - в базе данных сохраняется вся требуемая информация в едином формате, соответствующая условиям проверки. Использование информационной системы позволяет так же избавиться от ненужного дублирования данных, а, значит, и возможных ошибок.

Трудно спорить с тем фактом, что востребована только актуальная (соответствующая действительности) информация. В ситуации, когда разные сотрудники ведут собственные базы, частично дублирующие друг друга (например - база договоров у юристов и клиентская база у отдела продаж), возникает проблема поддержки информации в актуальном состоянии. Так при изменении, допустим, контактного телефона у клиента узнать об этом должны все заинтересованные подразделения компании. При использовании информационной системы такой проблемы попросту нет, изменившаяся информация доступна всем пользователям, имеющим соответствующие права. При использовании нескольких систем (например, системы бухгалтерского учета и системы планирования перевозок), задача актуализации информации решается еще на этапе разработки, таким образом, при изменении информации в одной системе, новые данные автоматически передаются в другую систему.

Информационная система характерна тем, что содержит так называемую бизнес-логику - правила и механизмы, которые регламентируют процессы работы с информацией. Таким образом, компания, внедрившая информационную систему, обеспечивает соблюдение единой технологии работы подразделений и филиалов. Все документы - будь то договор, накладная, коммерческое предложение или внутренние отчеты, выпускаемые разными сотрудниками при помощи информационной системы, оформлены единообразно в соответствии с фирменным стилем компании и внутренними требованиями.

В отношении корпоративной информации, задачами информационной безопасности в компании являются защита данных от хищения, искажения (умышленного или случайного), потери и ограничение доступа. Использование корпоративной информационной системы позволяет упростить решение этих задач - все данные хранятся в базе данных на сервере, ограничить физический доступ к серверу и хранящейся на нем информации гораздо проще и дешевле, чем решить ту же задачу для рабочих станций сотрудников. Сервера информационной системы подключены к источнику бесперебойного питания, данных хранятся на RAID-массивах (набор независимых дисков с частичным или полным дублированием хранящейся информации), обеспечивается регулярное резервное копирование данных на внешние носители. Все эти меры сводят практически к нулю риск потери данных, а если данные все же были утеряны, их можно оперативно восстановить с резервной копии. Данные, хранящиеся на сервере защищены от хищения - только системный администратор имеет доступ к серверу с максимальными правами. Остальные сотрудники обращаются к данным через интерфейс информационной системы, причем только к тем данным, доступ к которым разрешен выданными пользователю правами.

Для стратегического планирования, принятия управленческих решений, разбора спорных ситуаций необходимы исторические данные. Если в компании не внедрена корпоративная информационная система, уровень детализации исторических данных ограничивается, как правило, лишь историей выполненных заказов клиентов. Документы, сопровождающие выполнение заказов, конечно, тоже хранятся в архиве в твердой копии, но поиск и анализ такой неструктурированной информации представляет собой практически невыполнимую задачу. Внедрение корпоративной информационной системы с одной стороны решает ряд проблем, связанных с коллективной работой сотрудников в организации, и позволяет минимизировать финансовые и временные риски.

От данной ИС ожидается, что она позволит получению более рациональных вариантов решения управленческих задач за счет внедрения различных интеллектуальных систем, освобождению работников от различных рутинных работ за счет ее автоматизации, обеспечению достоверности информации, замене бумажных носителей данных на технологические для простоты дальнейшего доступа, что приводит к более рациональной организации переработки информации на компьютере и снижению объемов документов, уменьшению финансовых затрат, предоставлению потребителям уникальных и более производительных услуг, отысканию новых рыночных ниш, устранению различных проблем интересующихся в данной ИС потребителей.

Для создания данной ИС необходимо обеспечить найм программистов, для которых в задачу входит разбор предметной области, затем обеспечить им техническое задание, согласно которому нужно составить саму ИС. Составить руководство для программиста и для пользователя, а также документацию для дальнейшей возможности модификации и ее использования в системе.

Информация, которая хранится, обрабатывается и передается в ИС важна, т.к. она позовяет предприятию вести отчетность, ведь для работодателей учитывать рабочее время особенно важно, так как платить за простой подчиненных никто не хочет. Также и в законодательстве есть статья, которая рассказывает о важности учета рабочего времени и его оплаты, а КоАП предусматривает санкции в случае нарушения.

# **1.2 Описание входных и выходных данных**

При разборе темы «Учет и анализ рабочего времени» были выделены входные и выходные данные. Ниже представлены таблицы, каждая из которых отображает все необходимые данные.

Таблица 1.1 – Входные данные авторизации пользователя

|  |  |  |
| --- | --- | --- |
| Обозначение | Наименование | Тип |
|  | E-mail | Строка |
|  | Пароль | Строка |

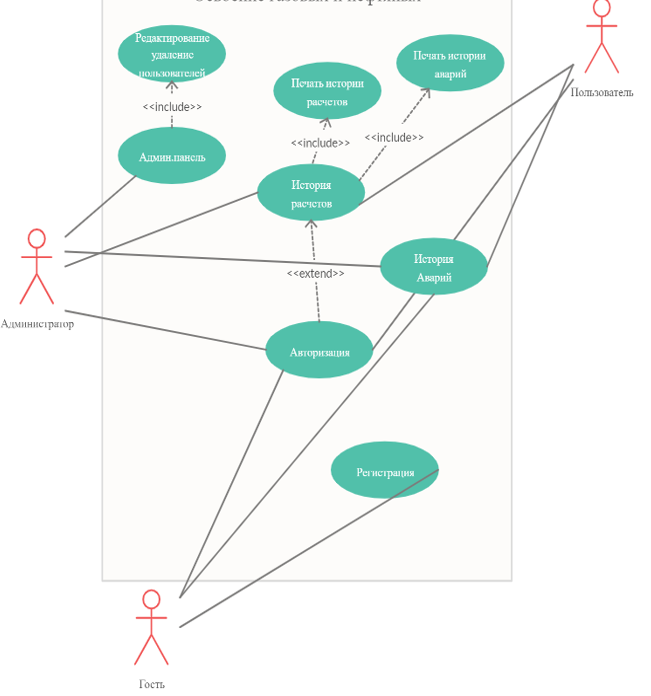
Таблица 1.2 – Входные данные регистрации пользователя

|  |  |  |
| --- | --- | --- |
| Обозначение | Наименование | Тип |
| id\_sotr | Идентификатор аккаунта | Число |
| Surname | Фамилия | Строка |
| Name | Имя | Строка |
| Lastname | Отчество | Строка |
| Phone | Телефон | Строка |
| Email | E-mail пользователя | Строка |
| Password | Пароль | Строка |
| Position | Должность | Строка |
| Datareg | Дата регистрации | Дата |
| namesector | Наименование отдела | Строка |

Таблица 1.3 – Выходные данные пользователя

|  |  |  |
| --- | --- | --- |
| Обозначение | Наименование | Тип |
| id\_sector | Идентификатор отдела | Строка |
| id\_sotr | Идентификатор аккаунта | Число |
| datenow | Текущая дата | Дата |
| datestart | Начало работы | Дата |
| datebreak | Начало перерыва | Дата |
| datebreakend | Конец перерыва | Дата |
| dateend | Конец работы | Дата |

# **1.3 Проектирование диаграммы прецедентов и схемы данных**



(Рисунок 1 – Диаграмма прецедентов)

Диаграмма вариантов использования (англ. use case diagram) в UML — диаграмма, отражающая отношения между акторами и прецедентами и являющаяся составной частью модели прецедентов, позволяющей описать систему на концептуальном уровне.

Прецедент — возможность моделируемой системы (часть её функциональности), благодаря которой пользователь может получить конкретный, измеримый и нужный ему результат. Прецедент соответствует отдельному сервису системы, определяет один из вариантов её использования и описывает типичный способ взаимодействия пользователя с системой. Варианты использования обычно применяются для спецификации внешних требований к системе.

Основное назначение диаграммы — описание функциональности и поведения, позволяющее заказчику, конечному пользователю и разработчику совместно обсуждать проектируемую или существующую систему.

Диаграммы прецедентов представляют собой один из пяти типов диаграмм, применяемых в UML для моделирования динамических аспектов системы (остальные четыре типа - это диаграммы деятельности, состояний, последовательностей и кооперации). Диаграммы прецедентов играют основную роль в моделировании поведения системы, подсистемы или класса. Каждая такая диаграмма показывает множество прецедентов, актеров и отношения между ними. Диаграммы прецедентов применяются для моделировании вида системы с точки зрения прецедентов (или вариантов использования). Чаще всего это предполагает моделирование контекста системы, подсистемы или класса либо моделирование требований, предъявляемых к поведению указанных элементов. Диаграммы прецедентов имеют большое значение для визуализации, специфицирования и документирования поведения элемента. Они облегчают понимание систем, подсистем или классов, представляя взгляд извне на то, как данные элементы могут быть использованы в соответствующем контексте. Кроме того, такие диаграммы важны для тестирования исполняемых систем в процессе прямого проектирования и для понимания их внутреннего устройства при обратном проектировании

При моделировании системы с помощью диаграммы прецедентов системный аналитик стремится:

* чётко отделить систему от её окружения;
* определить действующих лиц (актёров), их взаимодействие с системой и ожидаемую функциональность системы;
* определить в глоссарии предметной области понятия, относящиеся к детальному описанию функциональности системы (то есть прецедентов).

Работа над диаграммой может начаться с текстового описания, полученного при работе с заказчиком. При этом нефункциональные требования (например, конкретный язык или система программирования) при составлении модели прецедентов опускаются (для них составляется другой документ).

Для отражения модели прецедентов на диаграмме используются:

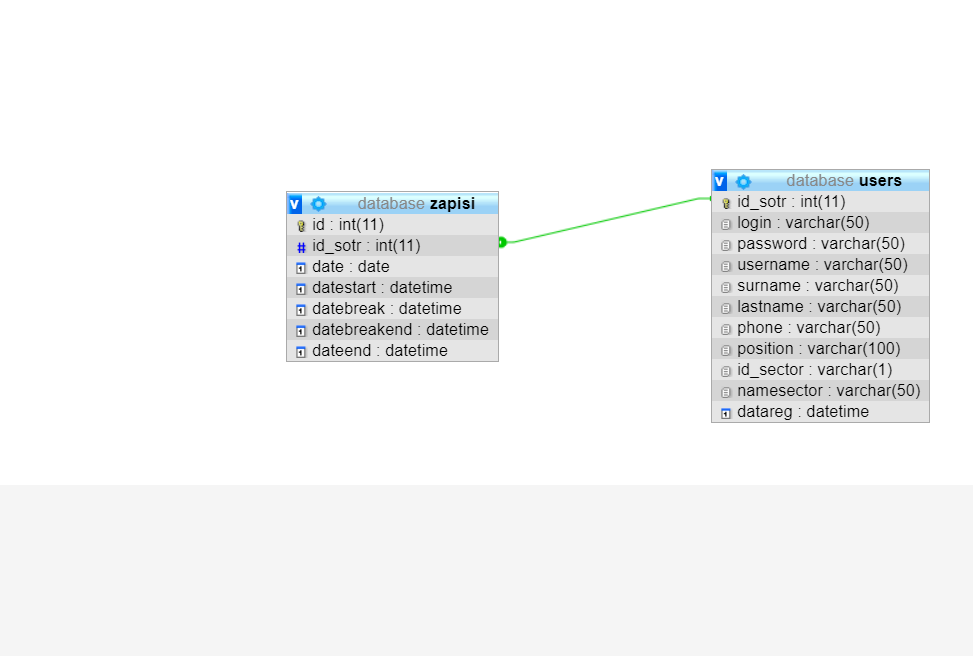
* рамки системы (англ. system boundary) — прямоугольник с названием в верхней части и эллипсами (прецедентами) внутри. Часто может быть опущен без потери полезной информации,
* актор (англ. actor) — стилизованный человечек, обозначающий набор ролей пользователя (понимается в широком смысле: человек, внешняя сущность, класс, другая система), взаимодействующего с некоторой сущностью (системой, подсистемой, классом). Акторы не могут быть связаны друг с другом (за исключением отношений обобщения/наследования),
* прецедент — эллипс с надписью, обозначающий выполняемые системой действия (могут включать возможные варианты), приводящие к наблюдаемым акторами результатам. Надпись может быть именем или описанием (с точки зрения акторов) того, «что» делает система (а не «как»). Имя прецедента связано с непрерываемым (атомарным) сценарием — конкретной последовательностью действий, иллюстрирующей поведение. В ходе сценария акторы обмениваются с системой сообщениями. Сценарий может быть приведён на диаграмме прецедентов в виде UML-комментария. С одним прецедентом может быть связано несколько различных сценариев.

Часть дублирующейся информации в модели прецедентов можно устранить указанием связей между прецедентами[[1]](https://ru.wikipedia.org/wiki/%D0%94%D0%B8%D0%B0%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B0_%D0%BF%D1%80%D0%B5%D1%86%D0%B5%D0%B4%D0%B5%D0%BD%D1%82%D0%BE%D0%B2#cite_note-%D0%91%D0%B0%D0%B1%D0%B8%D1%87-1):

* обобщение прецедента — стрелка с незакрашенным треугольником (треугольник ставится у более общего прецедента),
* включение прецедента — пунктирная стрелка со стереотипом «include»,
* расширение прецедента — пунктирная стрелка со стереотипом «extend» (стрелка входит в расширяемый прецедент, в дополнительном разделе которого может быть указана точка расширения и, возможно в виде комментария, условие расширения)

При работе с вариантами использования важно помнить несколько простых правил:

* каждый прецедент относится как минимум к одному действующему лицу;
* каждый прецедент имеет инициатора;
* каждый прецедент приводит к соответствующему результату.



(Рисунок 2 – схема данных системы)

Схема базы данных включает в себя описания содержания, структуры и ограничений целостности, используемые для создания и поддержки базы данных.

Постоянные данные в среде базы данных включают в себя схему и базу данных. Система управления базами данных (СУБД) использует определения данных в схеме для обеспечения доступа и управления доступом к данным в базе данных.

Схема базы данных (от англ. Database schema) — её структура, описанная на формальном языке, поддерживаемом СУБД. В реляционных базах данных схема определяет таблицы, поля в каждой таблице (обычно с указанием их названия, типа, обязательности), и ограничения целостности (первичный, потенциальные и внешние ключи и другие ограничения).

Схемы в общем случае хранятся в словаре данных. Хотя схема определена на языке базы данных в виде текста, термин часто используется для обозначения графического представления структуры базы данных.

Основными объектами графического представления схемы являются таблицы и связи, определяемые внешними ключами.

Схема (SCHEMA) является одним из основных объектов базы данных Oracle Database. Близкое понятие (RIS Schema) существует в RIS-интерфейсе доступа к базам данных. SCHEMA также появилась и в Microsoft SQL Server 2005 и формально определяется как набор объектов в базе данных.

В Oracle схема привязывается только к одному пользователю (USER) и является логическим набором объектов базы данных. Схема создаётся при [создании](https://ru.wikipedia.org/w/index.php?title=Create_(SQL)&action=edit&redlink=1) пользователем первого объекта, и все последующие объекты, созданные этим пользователем, становятся частью этой схемы.

Схема может включать другие объекты, принадлежащие этому пользователю:

* таблицы,
* последовательности,
* хранимые программы,
* кластеры,
* связи баз данных,
* триггеры,
* библиотеки внешних процедур,
* индексы,
* пакеты,
* хранимые функции и процедуры,
* синонимы,
* представления,
* снимки,
* объектные таблицы,
* объектные типы,
* объектные представления.

Существуют и подобъекты схемы, такие как:

* столбцы: таблиц и представлений,
* секции таблиц,
* ограничения целостности,
* триггеры,
* пакетные процедуры и функции и другие элементы, хранимые в пакетах (курсоры, типы и т. п).

Существуют объекты, независимые от схемы:

* каталоги,
* профили,
* роли,
* сегменты,
* табличные области,
* пользователи.

# **1.4 Выбор программных и технических средств разработки**

Данная информационная система разработана при помощи PHP, HTML, CSS, JavaScript, MySQL, Bootstrap 4.

PHP — 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.

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.

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.

CSS (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.

JavaScript — мультипарадигменный язык программирования. Поддерживает объектно-ориентированный, императивный и функциональный стили. Является реализацией языка ECMAScript (стандарт ECMA-262). JavaScript обычно используется как встраиваемый язык для программного доступа к объектам приложений. Наиболее широкое применение находит в браузерах как язык сценариев для придания интерактивности веб-страницам. Основные архитектурные черты: динамическая типизация, слабая типизация, автоматическое управление памятью, прототипное программирование, функции как объекты первого класса.

Bootstrap — свободный набор инструментов для создания [сайтов](https://ru.wikipedia.org/wiki/%D0%A1%D0%B0%D0%B9%D1%82) и [веб-приложений](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D0%BF%D1%80%D0%B8%D0%BB%D0%BE%D0%B6%D0%B5%D0%BD%D0%B8%D0%B5). Включает в себя [HTML](https://ru.wikipedia.org/wiki/HTML)- и [CSS](https://ru.wikipedia.org/wiki/CSS)-шаблоны оформления для [типографики](https://ru.wikipedia.org/wiki/%D0%A2%D0%B8%D0%BF%D0%BE%D0%B3%D1%80%D0%B0%D1%84%D0%B8%D0%BA%D0%B0), веб-форм, кнопок, меток, блоков навигации и прочих компонентов веб-интерфейса, включая [JavaScript](https://ru.wikipedia.org/wiki/JavaScript)-расширения. Преимущество его использования в его адаптивной сетке.

Рекомендуемое разрешение экрана для просмотра сайта от 1280 x 800 и выше. Электронная версия сайта воспроизводится на компьютерах с системным обеспечением Windows XP/7/8/10. Для просмотра сайта необходимо наличие одного из браузеров:

- Internet Explorer 8 и выше;

- Opera 10.0 и выше;

- Google Chrome 18.0 и выше;

- Mozilla Firefox 13.0 и выше.

# **2. Разработка программных решений**

При разработке базы данных нужно воспользоваться системой управления базами данных PhpMyAdmin. Для начала разработки приложения нужно скачать OpenServer. После установки программы, настроим ее, выставляем следующие на стройки как на рисунке 2.1:

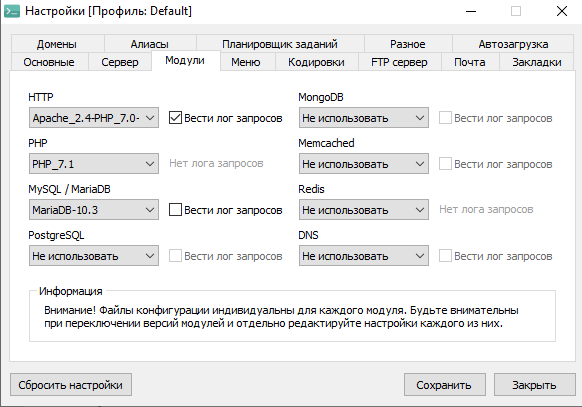


Рисунок 2.1 – Настройка OpenServer

Далее создадим наш проект в папке domains и запустим OpenServer. В созданной папке создаем следующую структуру проекта, показанную на рисунке 2.2:

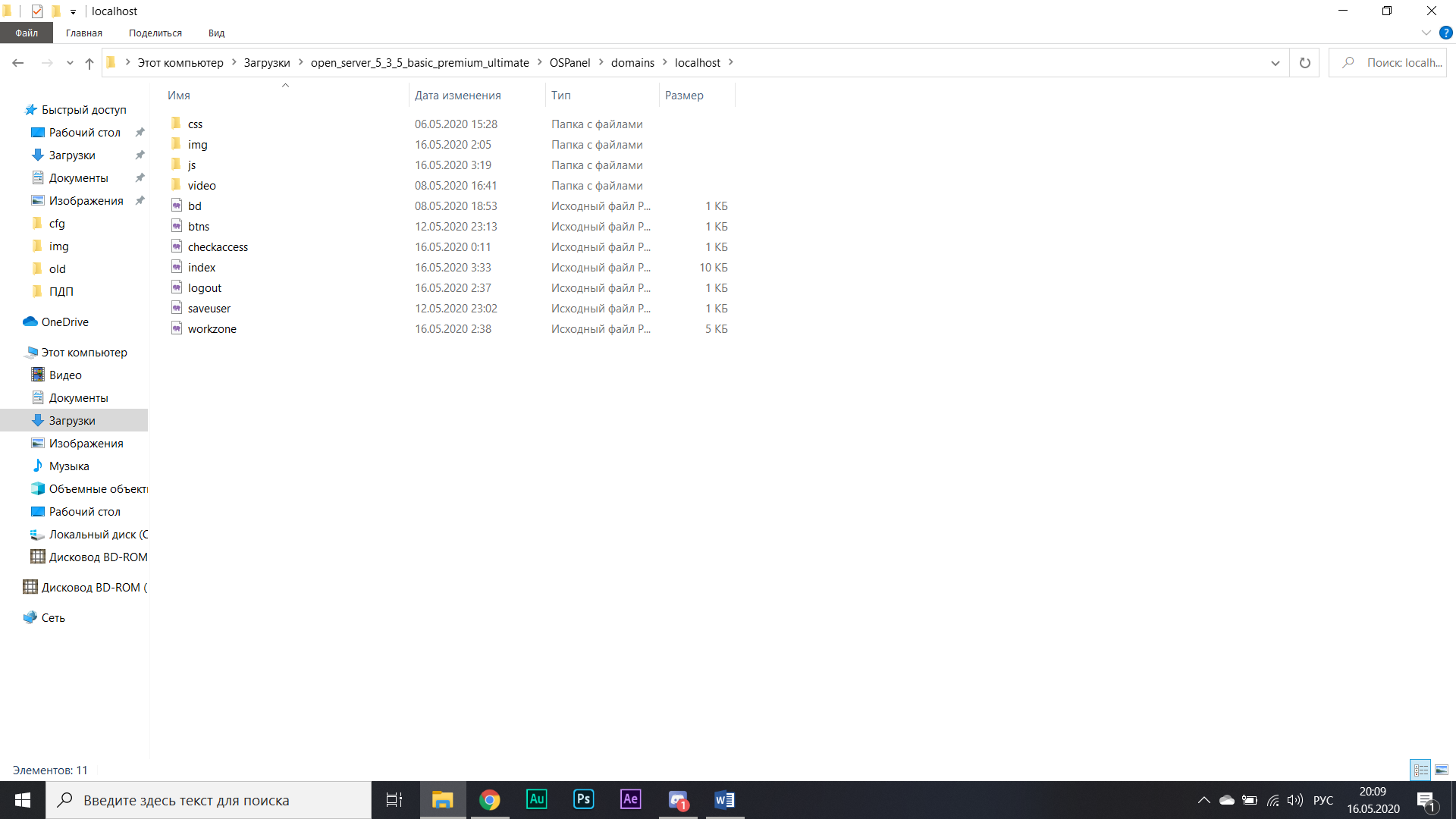


Рис 2.2 - Структура проекта

Настроим базу данных в файле bd.php как показано на рисунке 2.3:

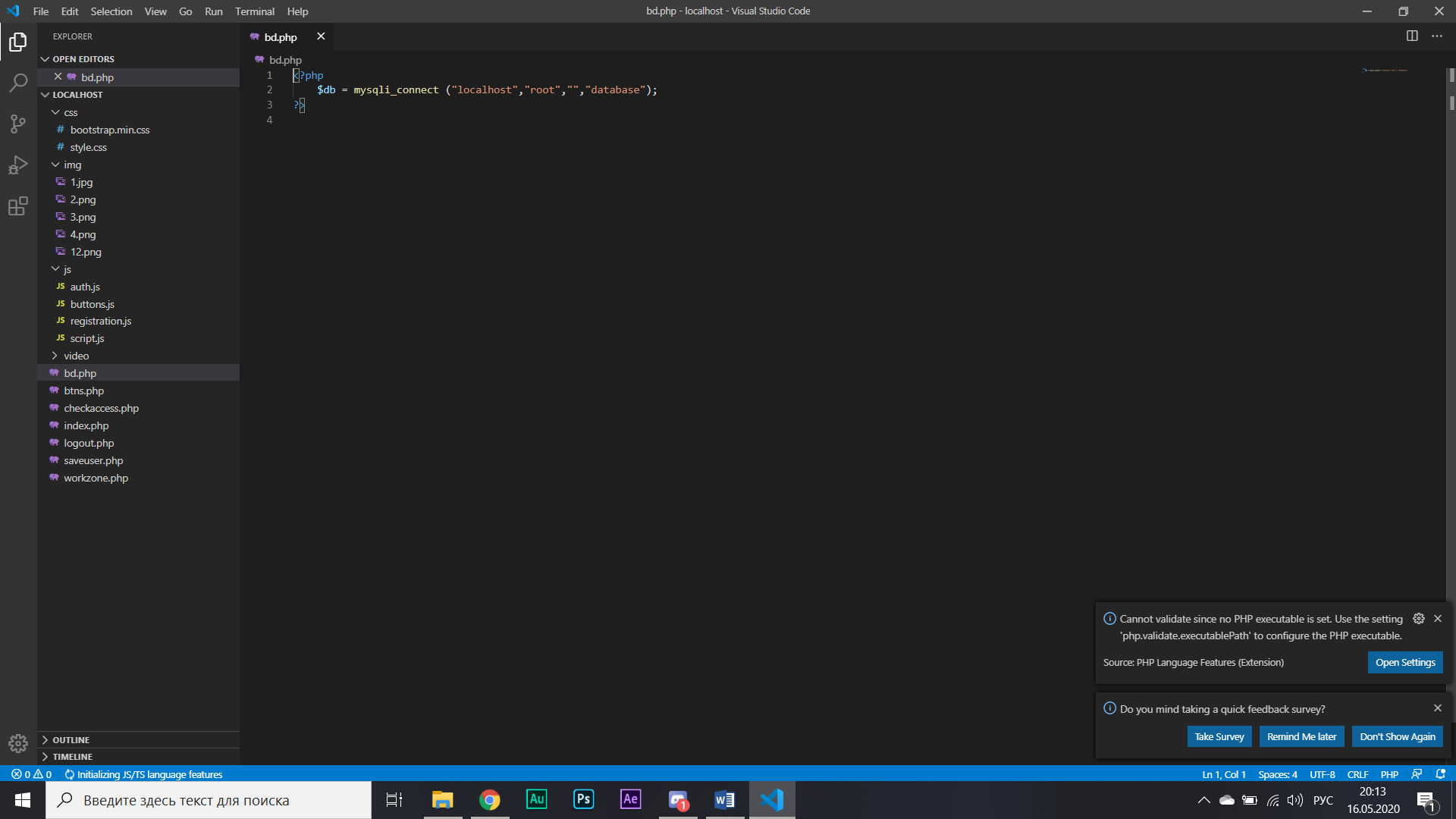


Рис 2.3 – Настройка файла bd.php

Чтобы разработать основной интерфейс, были использованы средства HTML, CSS, JS. Далее запускаем OpenServer и прописываем localhost/ в адресной строке. Видим стартовую страницу сайта.



Рис 2.4 – Стартовая страница index.php

Далее необходимо производится регистрация.

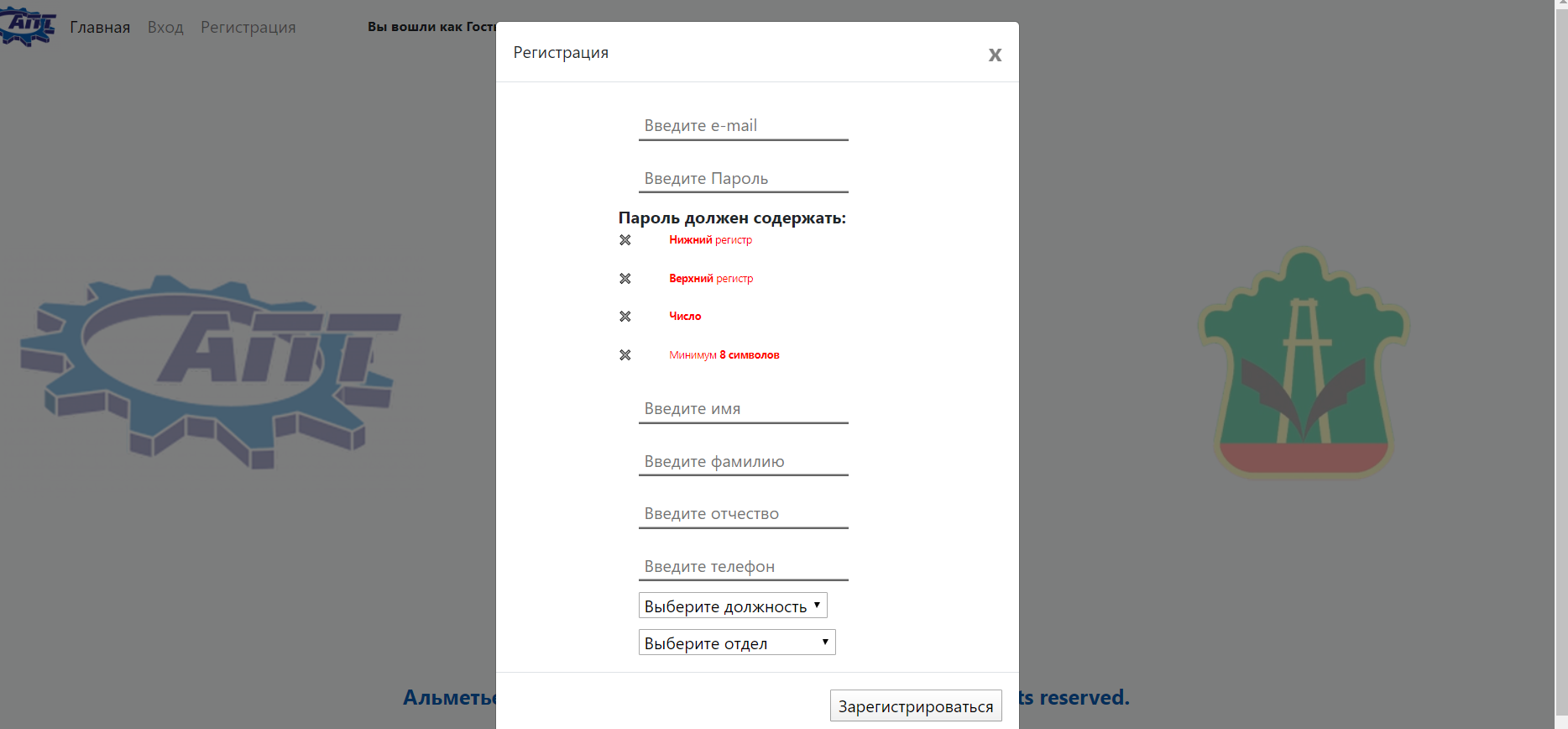


Рис 2.5 – Регистрация.

После чего производится вход на сайт, где доступен весь функционал системы.

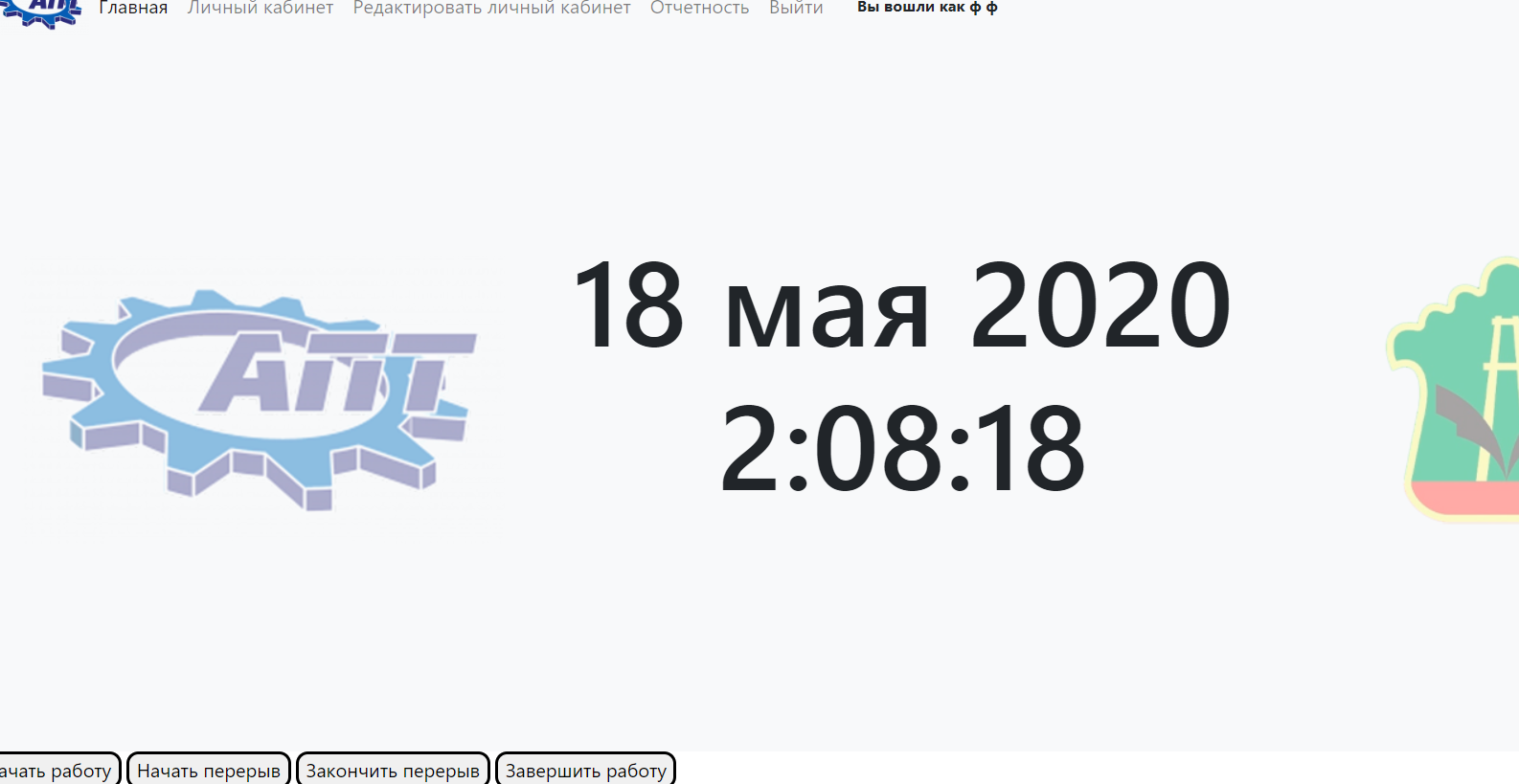


Рис 2.6 – Авторизованный пользователь в системе.

# **3. Тестирование программного продукта.**

Тестирование – процесс испытания программного продукта, ставящий своей целью проверку соответствия между реальным поведением программы и её ожидаемым поведением на конечном наборе тестов.

Целью тестирования является:

1. Предоставить заказчику готовый программный продукт, соответствующий установленным требованиям;
2. Выявить ситуации, в которых продукт будет вести себя некорректно, то есть не соответствуя заявленным требованиям.

Уровни тестирования:

1. Модульное тестирование (Автономное). На этом уровне отдельные элементы системы должны тестироваться максимально независимо от других элементов и, одновременно, быть пригодными для тестирования.
2. Комплексное тестирование (Сборочное тестирование, integration testing или interface testing). На этом уровне объединяют некоторую группу элементов системы и тестируют их совместно.
3. Системное тестирование (system testing). На этом уровне тестируется весь программный модуль или вся система целиком.
4. Приемочное тестирование (Приемо-сдаточное тестирование или acceptance testing). На этом уровне завершенный модуль тестируется Заказчиком, конечными пользователями или соответствующими уполномоченными с целью определения соответствия системы – требованиям Заказчика и готовности системы к внедрению.
5. Операционное тестирование (Release Testing). Если система удовлетворяет всем предъявленным требованиям, нужно убедиться, что она удовлетворит нужды конечного пользователя и выполнит отведенную ей роль в среде своей эксплуатации, как это было определено в бизнес модели системы.

# **Заключение**

В период прохождения преддипломной практики мною были выполнены следующие виды работ:

* анализ предметной области;
* изучение языка программирования JS, PHP;
* создание системы по учету и анализу рабочего времени;

Использование информационных технологий позволяет сократить время, требуемое конечному пользователю для выполнения конкретных производственных задач, уменьшить затраты при их реализации, исключить возможность появления ошибок.

За время выполнения данной преддипломной практики была достигнута поставленная цель – систематизированы и расширены знания по специальности, получены практические навыки, которые в будущем будут востребованы при решении конкретных задач.

# **Список использованных источников**

1. Экономическая информатика / Под ред. П.В.Конюховского и Д.Н.Колесова. – СПб: Питер, 2001. – 560с.: ил.

2. Кощеева Е.Л. Создание и использование музейных информационных ресурсов // Музей будущего: информационный менеджмент / Сост. А.В.Лебедев. М.: Прогресс-Традиция, 2009. - С.35-45. <http://future.museum.ru/part01/010205.htm>