МИНИСТЕРСТВО ОБРАЗОВАНИЯ и науки РЕСПУБЛИКИ БАШКОРТОСТАН

Государственное автономное ПРОФЕССИОНАЛЬНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

УФИМСКИЙ КОЛЛЕДЖ СТАТИСТИКИ, ИНФОРМАТИКИ И ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ

ОТЧЕТ ПО ИНДИВИДУАЛЬНОМУ ПРОЕКТУ

На тему «Фриланс Биржа»

По УП 09.01 Учебная практика

Выполнил студент группы 20ВЕБ1

Сафиуллин Азамат Румилевич

Проверил(а) преподаватель информатики

и программирования

Дмитриева Елизавета Константиновн

2023

СОДЕРЖАНИЕ

|  |  |
| --- | --- |
|  | лист |
| 1 Описание предметной области | 3 |
| 2 Описание входной информации | 4 |
| 3 Описание выходной информации  4 UML диаграммы | 4  5 |
| 5 Логическое моделирование | 5 |
| 6 Описание структуры базы данных | 5 |
| 7 Контрольный пример | 8 |
| 8 Общие требования к программному продукту  9 Требования к навигации и взаимодействию пользователя с системой. | 9  10 |
| 10 Карта сайта | 12 |
| 11 Физическая схема | 12 |
| 12 Описание модулей | 13 |
| 13 Протоколы тестирования  14 Руководство пользователя | 14  20 |
| Приложение А. UML диаграммы | 27 |
| Приложение Б. Даталогическая модель БД  Приложение В. Физическая схема сайта | 28  29 |
| Список литературы | 30 |

**1. Описание предметной области**

Такие «Фриланс биржи» являются разновидностью веб-сайтов, функционал которого, гораздо больше и навигация удобнее. Данный вид сайта ориентированы на определенную тематику и объединяют вокург себя пользователей с одинаковыми интересами.

Портал для фрилансеров – сайт, предоставляющий возможность удаленной работы разработчикам, дизайнерам и другим специалистам, так же является посредником между фрилансером и клиентом.

Пройдя регистрацию пользователь уже может, воспользовавшись услугой сайта по размещению контента, выставив задачу на сайт или стать исполнителем.

Выставленные задачи на сайте и вся информация хранится в базе данных. Потенциальный клиент, зайдя на полную информацию заказа, может узнать о предоставленных услугах исполнителем так и о самом разработчике. После выбора задачи, удовлетворяющие все условия пользователя, он может оставить заявку на выполнение заказа, с предоставлением всей информации, технического задания, заложить сроки выполнения проекта, и сумму выделенную на выполнение задачи (система оплаты не будет реализована). Для наглядности процесса выполнения задачи, будет реализован функционал вывода статусов заказа.

Сайт создается с нуля на php фреймворке laravel, без использования дополнительных систем управления по типу bitrix, WordPress и т.д.

Существуют огромное количество проектов по бирже фриланса, которые на столько закрепились на данной позиции, что при одном только упоминаний о фрилансе, их сайты появляются в поисковых результатах.

Взяв во внимание и проанализировав существующие сайты, можно выделить следующий основной функционал, который так же будет использоваться в данном проекте:

* Регистрация и авторизация;
* Просмотр локальной информации по теме портала;
* Простая и удобная навигация по сайту;
* Создание задач и заказов для исполнителей;
* Просмотр задачи и заказов;
* Возможность писать отзывы к заказам.

1. **Описание входной информации**

Входными данными будут, задачи от заказчиков, предоставленные на выполнение исполнителям.

Таблица 1.2.1 – описание входных документов

|  |  |  |
| --- | --- | --- |
| Наименование документа | Дата поступления документа | Откуда поступает документ |
| Заказ для исполнителя | При подаче | Клиент |

1. **Описание выходной информации**

Выходными данными будут, количество новых заказов в месяц, количество выполненных заказов в месяц, количество зарегестрированных исполнителей.

Табица 1.3.1 – описание выходных документов

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Наименование документа | Периодичность выдачи документа | Кол-во экз. | Куда передаются | Итоги |
| Документ о заказе | Каждый раз при заказе | 1 | Клиенту | Информация о заказе |
| Отчёт о заказах | Каждый месяц | 1 | Администратору на почту | Кол-во новых заказов за месяц, кол-во выполненных заказов за месяц |

1. **Use Case Диаграмма (диаграмма прецедентов)**

Диаграмма прецедентов – это диаграмма, которая позволяет  
визуализировать различные типы ролей и их взаимодействия с системой.

В данной диаграмме существует 4 объекта:  
- актер;  
- случай использования;  
- система;  
- пакет.

Актер в диаграмме прецедентов – это любая сущность, которая  
взаимодействует с системой. Это может быть человек, организация или  
внешняя система.

Случай использования представляет собой действие внутри системы.

Система используется для определения сферы применения.

Пакет является дополнительным параметром, который используется в  
сложных диаграммах. Пакеты используются для группировки большого  
количества случаев использования.

Каждая группа пользователей на диаграмме вариантов использования  
изображается человечком.

1. **Логическая модель базы данных**

Под даталогической понимается модель, отражающая логические взаимосвязи между элементами данных безотносительно их содержания и физической организации.

Даталогическая модель представлена в приложении Б, рисунок Б.1

1. **Описание структуры базы данных**

Таблица 6.1 – User (Пользователи)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя поля | Описание поля | Тип данных | Размер поля | Тип ключа (РК-первичный, FK- внешний) |
| id | Id пользователя | integer | 5 | PK |
| name | ФИО пользователя | varchar | 100 |  |
| desctiption | Описание пользователя | varchar | 2000 |  |
| thumb | Фотография пользователя | varchar | 50 |  |
| category\_user | Роль пользователя | integer | 5 | FK |
| email | Электронная почта | varchar | 1000 |  |
| password | Пароль | varchar | 1000 |  |
| updated\_at | Дата изменения | datetime | 10 |  |
| created\_at | Дата создания | datetime | 10 |  |

Таблица 6.2 – Заявки (Orders)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя поля | Описание поля | Тип данных | Размер поля | Тип ключа (РК-первичный, FK- внешний) |
| id | Id заявки | integer | 5 | PK |
| titile | Навзание заявки | varchar | 100 |  |
| desctiption | Описание заявки | varchar | 2000 |  |
| files | Файлы заявки | varchar | 50 |  |
| client\_id | Id заказщика | integer | 5 | FK |
| categories | Категория заявки | integer | 5 | FK |
| status\_id | Статус заявки | integer | 5 | FK |
| price | Плата за выполнение заявки | datetime | 10 |  |
| Продолжение таблицы 1.2 | | | | |
| date | Сроки выполнения заявки | integer | 100 |  |

Таблица 6.3 – Заявка пользователь(OrderUser)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя поля | Описание поля | Тип данных | Размер поля | Тип ключа (РК-первичный, FK- внешний) |
| id | Id связи | integer | 5 | PK |
| user\_id | Id заказщика | integer | 5 | FK |
| order\_id | Id заявки | integer | 5 | FK |
| status\_id | Статус заявки | integer | 5 | FK |

Таблица 6.4 – Категория (Categories)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя поля | Описание поля | Тип данных | Размер поля | Тип ключа (РК-первичный, FK- внешний) |
| id | Id связи | integer | 5 | PK |
| slug | Символьный код категории | integer | 1000 |  |
| title | Название категории | integer | 1000 |  |

Таблица 6.4 – Роль (category\_user)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя поля | Описание поля | Тип данных | Размер поля | Тип ключа (РК-первичный, FK- внешний) |
| id | Id связи | integer | 5 | PK |
| slug | Символьный код категории | integer | 1000 |  |
| title | Название категории | integer | 1000 |  |

1. **Контрольный пример**

Таблица 7.1 – Пользователь

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ФИО | О себе | Фотография | Роль | Почта | Пароль |
| Азамат | Работник | Ссылка | Исполнитель | Azamsaf21@gmail.com | 123123123 |
| Олег | Работник | Ссылка | Исполнитель | saf21@gmail.com | 1234533123 |
| Игорь | Покупатель обычный | Ссылка | Клиент | Azaf21@gail.com | 1232425123 |

Таблица 7.2 - Роль

|  |
| --- |
| Название роли |
| Администратор |
| Исполнитель |
| Клиент |

Таблица 7.3 - Категории

|  |
| --- |
| Название |
| Дизайн |
| Текст и СЕО |
| Программирование |
| Соцсети и реклама |

Таблица 7.4 - Заявка пользователь

|  |  |  |
| --- | --- | --- |
| Id user | Id order | Id status |
| 5 | 2 | 3 |
| 7 | 5 | 1 |
| 3 | 5 | 1 |
| 8 | 4 | 5 |

Таблица 7.5 - Заявки

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Название | Описание | Файлы | Id клиента | Категория | Статус | Стоимость | Сроки |
| Создание лого | Создать уникальный логотип из 3 цветов компании, цвета предоставлю в тз | ТЗ.txt | 3 | Дизайн | В ожидании принятия | 12000 | 10 |

1. Общие требования к программному продукту

8.1 Введение

Краткое описание проекта

Working – это фриланс биржа, или простыми словами онлайн доска объявлений.  
Основным назначением проекта является работа с заявками от заказчиков для исполнителей.

Цели и задачи разработки

Working создается с целью:

-удобное размещение объявления на сайте;

-безопасное предоставление услуг;

Ссылки на связанные документы или проекты

https://github.com/Azazel1q/azamatlaravel/tree/main/documentation

8.2Функциональные требования

Подробное описание функциональности, которую должен

предоставлять программный продукт.

8.3 Входы и выходы системы.

Входы:

- заявка пользователя на выполнение для исполнителей;

Выходы:

- выполненная работа.

Взаимодействие с другими системами.

Отсутствует

Требования к интерфейсу

8.4 Описание пользовательского интерфейса.

8.1 главная страница:

- логотип компании;

-меню навигации;

-категории;

-быстрый переход на добавление заявки;

-вывод лучших исполнителей.

8.2 страница подачи заявки:

- поля для ввода данных;

- кнопка “Добавить”.

8.3 личный кабинет пользователя:

-информация о пользователе;

-выполненные заказы;

-все добавленные заявки;

9. Требования к навигации и взаимодействию пользователя с системой.

9.1 навигационное меню:

- фиксированное меню для быстрого доступа ко всем разделам.

9.2 просмотр категорий:

- возможность просмотра заявок клиентов по категориям.

9.3 подача объявления:

- заполнить соответствующие поля;.

9.4 личный кабинет:

-возможность просмотра информции о себе, о своих заявках и о выполненных заказах

9.5 адаптивность:

- отзывчивый дизайн для различных разрешений экрана.

9.6 аутентификация и профили:

- возможность регистрации и авторизации пользователя;

- личный кабинет.

9.7 уведомления:

- оповещение пользователя о вводе неправильных данных.

9.1 Требования к производительности

Ожидаемые показатели производительности.

Пропускная способность: 100;

Время ответа сервера: 2мс;

Время отображения: 2мс.

9.2 Требования к загрузке системы при различных нагрузках.

Запросов в секунду: не более 1000 в секунду;

Пользователей в системе:

Активных – 4 человек.

Зарегистрированных – 100 человек.

Одновременно работающих – 4 человек.

9.3 Требования к безопасности

Требования к управлению доступом и аутентификации.

Доступ к учетным записям администратора, мастера, менеджера и

клиента разграничивается c помощью механизма Laravel – Middleware.

Меры безопасности для защиты данных.

Пароли пользователей автоматически хешируются при регистрации.

9.4 Требования к обеспечению конфиденциальности.

Требования к обеспечению конфиденциальности основываются на:

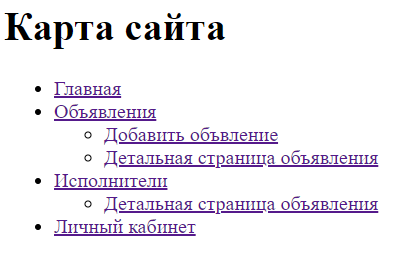
Закон РФ от 27.12.1991 N 2124-1(ред. от 13.06.2023)»О средствах массовой

информации»

10. Карта сайта

HTML карта – это статическая веб-страница, которая включает все основные и подстраницы вашего сайта в соответствии с их иерархией. Проще говоря, это что-то вроде оглавления книги, которое разработано для сайта, чтобы помочь посетителям легко находить страницы на вашем ресурсе.

Карта сайта представлена на рисунке 1.10.1.



1. Физическая схема

Схема базы данных — это структура и организация базы данных, которая определяет ее таблицы, поля, связи, ограничения и типы данных. Он служит образцом для организации данных и доступа к ним, предоставляя план действий для разработчиков, администраторов и пользователей при работе с базой данных. Схема необходима для эффективного и действенного управления данными, что приводит к повышению производительности базы данных и упрощению обслуживания.

Физическая схема базы данных представлена в приложении В.

12. Описание модулей

При запуске программного продукта выполняется проект «Working», после которого выполняются следующие процедуры, которые представлены в таблице 12.1.

Таблица 12.1 – Описание модулей

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Модуль | | Метод | | Назначение | | |
| UserController | | main | | Вывод главной страницы | | |
| workers | | Вывод исполнителей | | |
| detail | | Вывод детальной информации об исполнителей | | |
| Lk | | Вывод в личный кабинет информации о пользователе | | |
| OrderController | | index | | Вывод главной страницы объявлений | | |
| detail | | Вывод детальной информации об объявлений | | |
| orderForm | | Вывод формы для создания объявления | | |
| create | | Функция создания объявления | | |
| createOrderUser | | Функция создания связи между объявлением и пользователем | | |
| updateStatus | | Функция изменения статуса связи между объявлением и пользователем | | |
| delete | | Функция удаления объявления | | |
| CategoryController | | show | | Вывод детальной страницы категории с объявлениями этой категории | | |
| UserMiddleware.php | |  | | Разграничение доступа для клиента |

13. Протоколы тестирования

Тестирование программного продукта является неотъемлемой частью разработки системы. Его проводят для достижения наилучшей работоспособности и исправления ряда ошибок, которые могут возникнуть при взаимодействии с системой.

Обычно тестирование ПО проводится на всех этапах жизненного цикла: с момента разработки ПО до его запуска в эксплуатацию.

Цель тестирования - проверка соответствия ПО предъявляемым требованиям, обеспечение уверенности в качестве ПО, поиск очевидных ошибок в программном обеспечении.

Для выявления ошибок в работе системы были проведены следующие тесты.

Общая информация о тестировании приведена в таблице 13.1.

Таблица 13.1 – Общая информация о тестировании

|  |  |
| --- | --- |
| Общая информация о тестировании | |
| Название теста | Test\_system |
| Номер версии | 1.0 |
| Имя тестера | Сафиуллин Азамат |
| Дата тестирования | 21.12.2023 |

Тестирование авторизации при вводе некорректных данных приведено в таблице 13.2.

Таблица 13.2 - Тестирование авторизации при вводе некорректных данных

|  |  |
| --- | --- |
| Наименование | Описание |
| Test Case # | T\_AU\_1 |
| Приоритет тестирования | Высокий |
| Название тестирования | Проверка авторизации |
| Резюме испытания | Необходимо добиться корректного поведения системы, при вводе неправильных данных при авторизации пользователя |
| Шаги тестирования | 1. Открыть страницу авторизации пользователя; 2. Ввести в поля «Email» и «Пароль» неправильные данные; 3. Нажать на кнопку «Вход». |
| Данные тестирования | Логин: sdgsdg;  Пароль: sdgsdggsd. |
| Ожидаемый результат | Сайт должен вывести сообщение о неправильном вводе данных |
| Фактический результат | Вывелось сообщение «Неправильная почта или пароль». Результат на рисунке 13.2. |

|  |  |
| --- | --- |
| Предпосылки | Открытие сайта |
| Постусловия | Работа системы происходит корректно |
| Статус (Pass/Fall) | Pass |

Рисунок 13.2 – Ошибка при вводе некорректных данных авторизации

Таблица 13.3 - Тестирование авторизации при вводе корректных данных

|  |  |
| --- | --- |
| Наименование | Описание |
| Test Case # | T\_AU\_2 |
| Приоритет тестирования | Высокий |
| Название тестирования | Проверка авторизации с вводом корректных данных |
| Резюме испытания | Необходимо добиться корректного поведения системы, при вводе правильных данных при авторизации пользователя |
| Шаги тестирования | 1. Открыть страницу авторизации пользователя; 2. Ввести в поля «Email» и «Пароль» правильные данные; 3. Нажать на кнопку «Вход». |
| Данные тестирования | Пароль: admin@admin.ru;  Пароль: admin. |
| Ожидаемый результат | Сайт должен перенаправить пользователя на главную страницу. |

|  |  |
| --- | --- |
| Фактический результат | Пользователя перенаправило на главную страницу. |
| Предпосылки | Открытие сайта |
| Постусловия | Работа системы происходит корректно |
| Статус (Pass/Fall) | Pass |

Таблица 13.4 - Тестирование создания объявления с незаполненными полями ввода данных

|  |  |
| --- | --- |
| Наименование | Описание |
| Test Case # | T\_OR\_1 |
| Приоритет тестирования | Высокий |
| Название тестирования | Проверка авторизации в модуле OrderController.php с незаполненными полями ввода данных |
| Резюме испытания | Необходимо добиться корректного поведения системы, при незаполненных полях ввода данных |
| Шаги тестирования | 1. Авторизоваться в системе; 2. Перейти на вкладку «Объявления»; 3. Нажать на кнопку «Добавить объявление» 4. Оставить поля пустыми и нажать на кнопку добавить |
| Данные тестирования | - |
| Ожидаемый результат | Сайт должен вывести ошибку рядом с незаполненными полями |
| Фактический результат | Вывелись ошибку: «Поле пустое»  Результат на рисунке 13.4. |

|  |  |
| --- | --- |
| Предпосылки | Открытие сайта |
| Постусловия | Работа системы происходит корректно |
| Статус (Pass/Fall) | Pass |

Тестирование подачи заявки с заполненными полями ввода данных 13.5.

Таблица 13.5 - Тестирование создания объявления с заполненными полями ввода данных

|  |  |
| --- | --- |
| Наименование | Описание |
| Test Case # | T\_OR\_2 |
| Приоритет тестирования | Высокий |
| Название тестирования | Проверка авторизации в модуле OrderController.php с заполненными полями ввода данных |
| Резюме испытания | Необходимо добиться корректного поведения системы, при незаполненных полях ввода данных |

Продолжение таблицы 13.5

|  |  |
| --- | --- |
| Шаги тестирования | 1. Авторизоваться в системе; 2. Перейти на вкладку «Обявления» 3. Нажать на кнопку «Добавить объявление» 4. Заполнить поля по соответсвию |
| Данные тестирования | «Заполнить поля соответсвенно» |
| Ожидаемый результат | Сайт должен перенаправить пользователя на страницу с объявлениями |
| Фактический результат | Сайт перенаправил пользователя |
| Предпосылки | Открытие сайта |
| Постусловия | Работа системы происходит корректно |
| Статус (Pass/Fall) | Pass |

Таблица 13.6 - Тестирование регистрации пользователя с вводом некорректных данных

|  |  |
| --- | --- |
| Наименование | Описание |
| Test Case # | T\_REG\_1 |
| Приоритет тестирования | Высокий |
| Название тестирования | Проверка регистрации с вводом некорректных данных |
| Резюме испытания | Необходимо добиться корректного поведения системы, при вводе некорректных данных в регистрации |

|  |  |
| --- | --- |
| Шаги тестирования | 1. Перейти на вкладку «Регистрация»; 2. Ввести в поля некорректные данные; 3. Нажать на кнопку «Зарегистрироваться». |
| Данные тестирования | Имя: враповл;  Email: trythgf;  Пароль: 12312;  Выбор исполнителя: Клиент. |
| Ожидаемый результат | Сайт должен вывести ошибки о вводе некорректных данных |
| Фактический результат | Сайт вывел ошибку «Незаполнены поля»  Результат на рисунке 13.5. |
| Предпосылки | Открытие сайта |
| Постусловия | Работа системы происходит корректно |
| Статус (Pass/Fall) | Pass |

Рисунок 13.5 – Ошибка при вводе некорректных данных регистрации

Тестирование регистрации пользователя с вводом корректных данных 13.7.

Таблица 13.7 - Тестирование регистрации пользователя с вводом корректных данных

|  |  |
| --- | --- |
| Наименование | Описание |
| Test Case # | T\_REG\_2 |
| Приоритет тестирования | Высокий |
| Название тестирования | Проверка регистрации в модуле с вводом корректных данных |
| Резюме испытания | Необходимо добиться корректного поведения системы, при вводе корректных данных в регистрации |
| Шаги тестирования | 1. Перейти на вкладку «Регистрация»; 2. Ввести в поля корректные данные; 3. Нажать на кнопку «Зарегистрироваться». |
| Данные тестирования | Имя: Анатолий;  Фамилия: Петров;  Почта: anat@yandex.ru;  Пароль: Fyfnjkbq123. |
| Ожидаемый результат | Сайт должен перенаправить пользователя на главную страницу и автоматически авторизовать его |
| Фактический результат | Сайт перенаправил пользователя на главную страницу и авторизовал его |
| Предпосылки | Открытие сайта |
| Постусловия | Работа системы происходит корректно |
| Статус (Pass/Fall) | Pass |

Тестирование добавления нового пользователя администратором с вводом некорректных данных 13.8.

14 Руководство пользователя

Главной целью руководства пользователя является обучение работы с системой.

Данный программный продукт предназначен для просмотра и изменения информации, которая хранится в базе данных, а точнее регистрации нового пользователя, добавления объявления, изменения статуса объявления, выбор объявления и предоставления своей кандидатуры.

Минимальные требования к аппаратному обеспечению для производительной работы программного продукта:

* операционная система Windows 10;
* разрядность системы: x64;
* ОЗУ: от 2Гб.;
* монитор;
* клавиатура;
* компьютерная мышь.

Для запуска и комфортного использования программного продукта пользователю необходимы базовые знания работы с браузером и операционной системой Windows.

Подготовка системы к работе:

* перейти на страницу сайта по его URL.

После перехода по URL, пользователя перекидывает на главную страницу (рисунок 13.1).

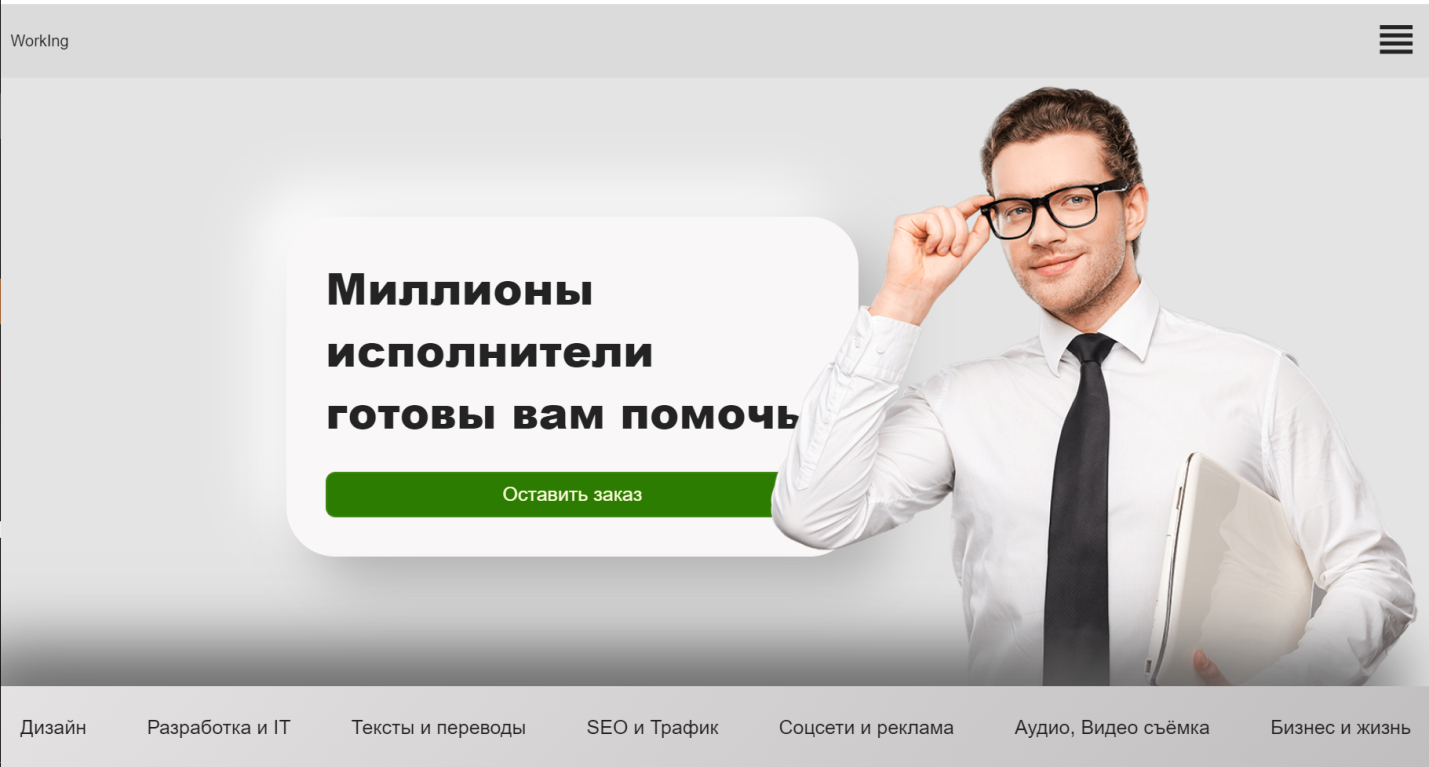


Рисунок 13.1 – Главная страница

Нажав на кнопку оставить заявку, пользователя перенаправляет на страницу авторизации, после успешной авторизации пользователь может добавить новое объявление.

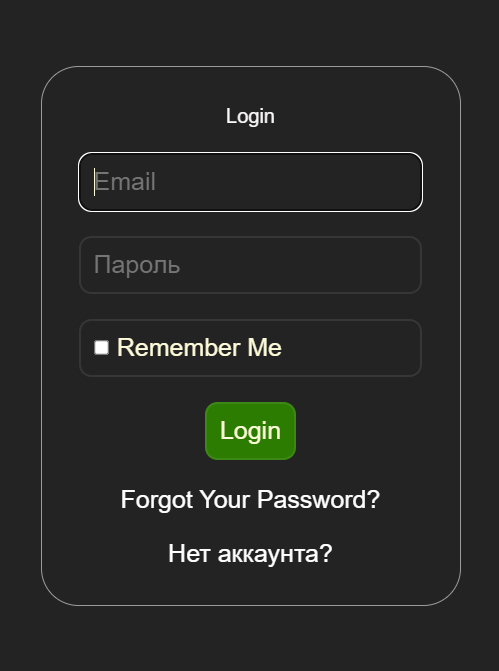


Рисунок 13.2 – Авторизация пользователя

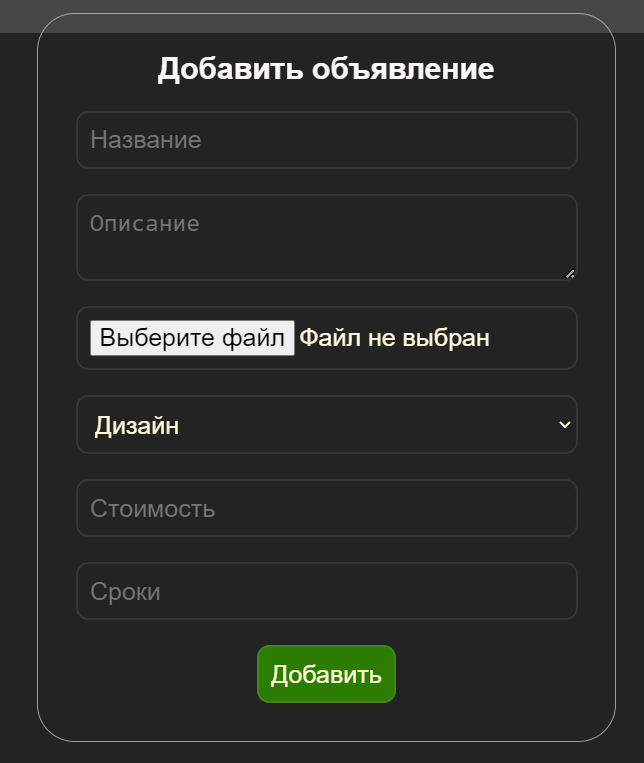


Рисунок 13.3 – форма добавления объявления.

После этого пользователь может ждать, либо просматривать исполнителей, если пользователь открыв меню перейдет на страницу «Исполнители». Так же есть возможность просматривать профили исполнителей.

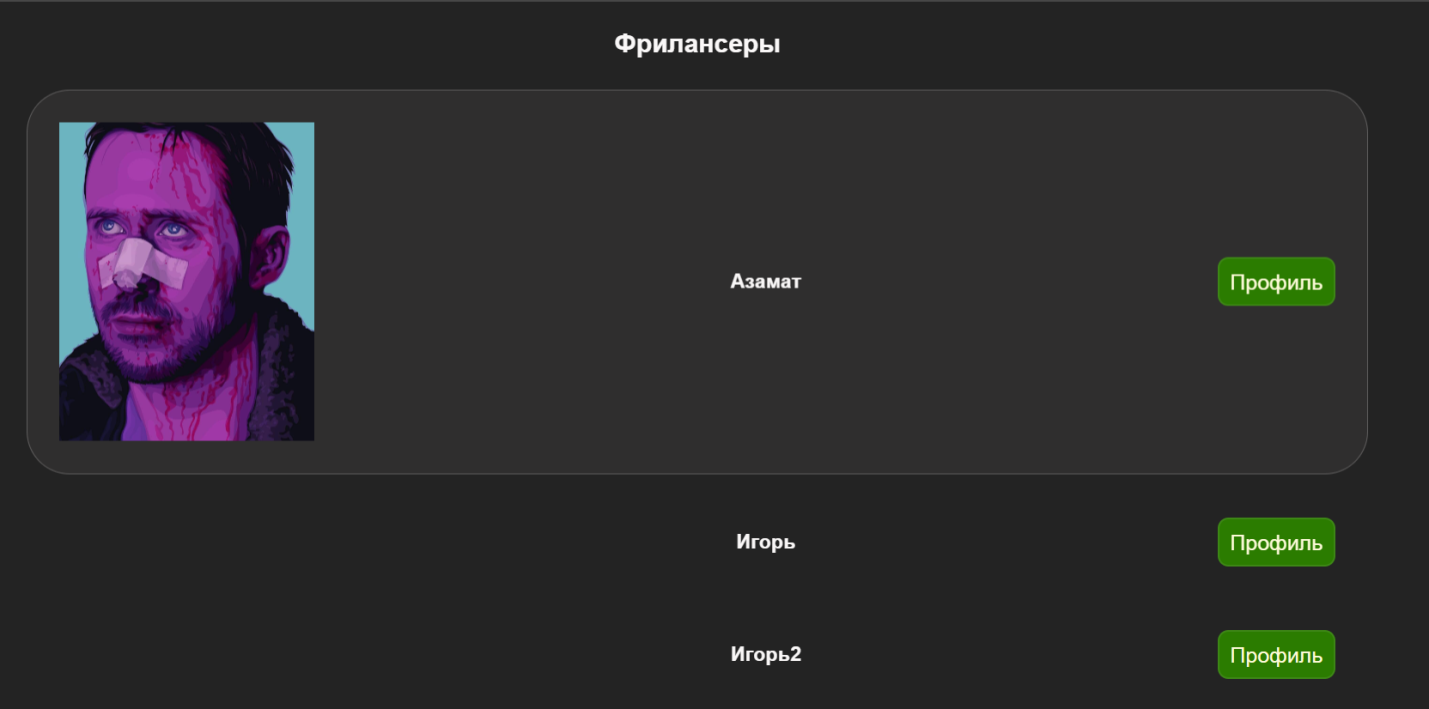


Рисунок 13.4 – страница «Исполнители»

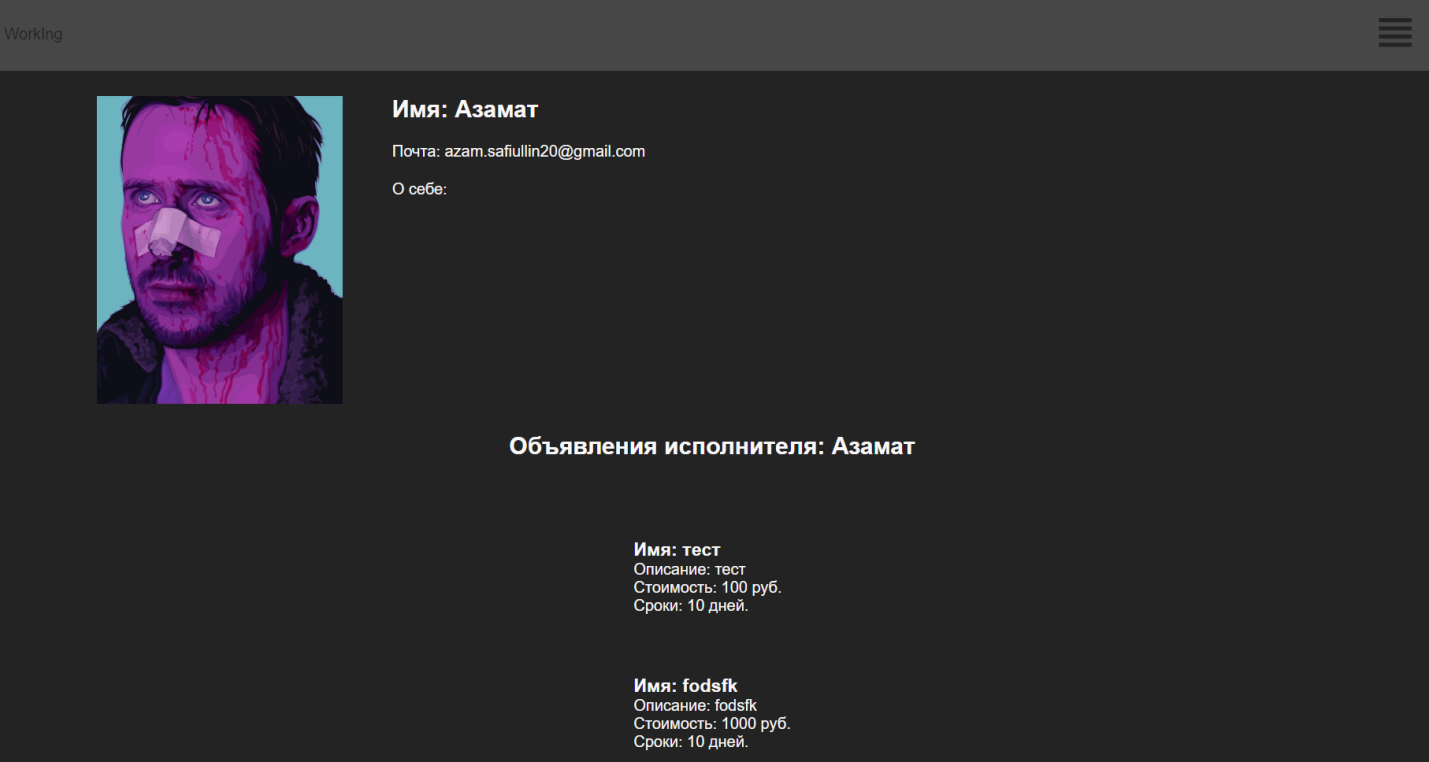


Рисунок 13.5 – профиль исполнителя

Так же пользователь может просматривать свои объявления, перейдя на страницу личного кабинета, если пользователь при регистрации выбрал роль исполнитель, то дополнительно в личном кабинете появится заказ, которые он предложил свою кандидатуру на выполнение.

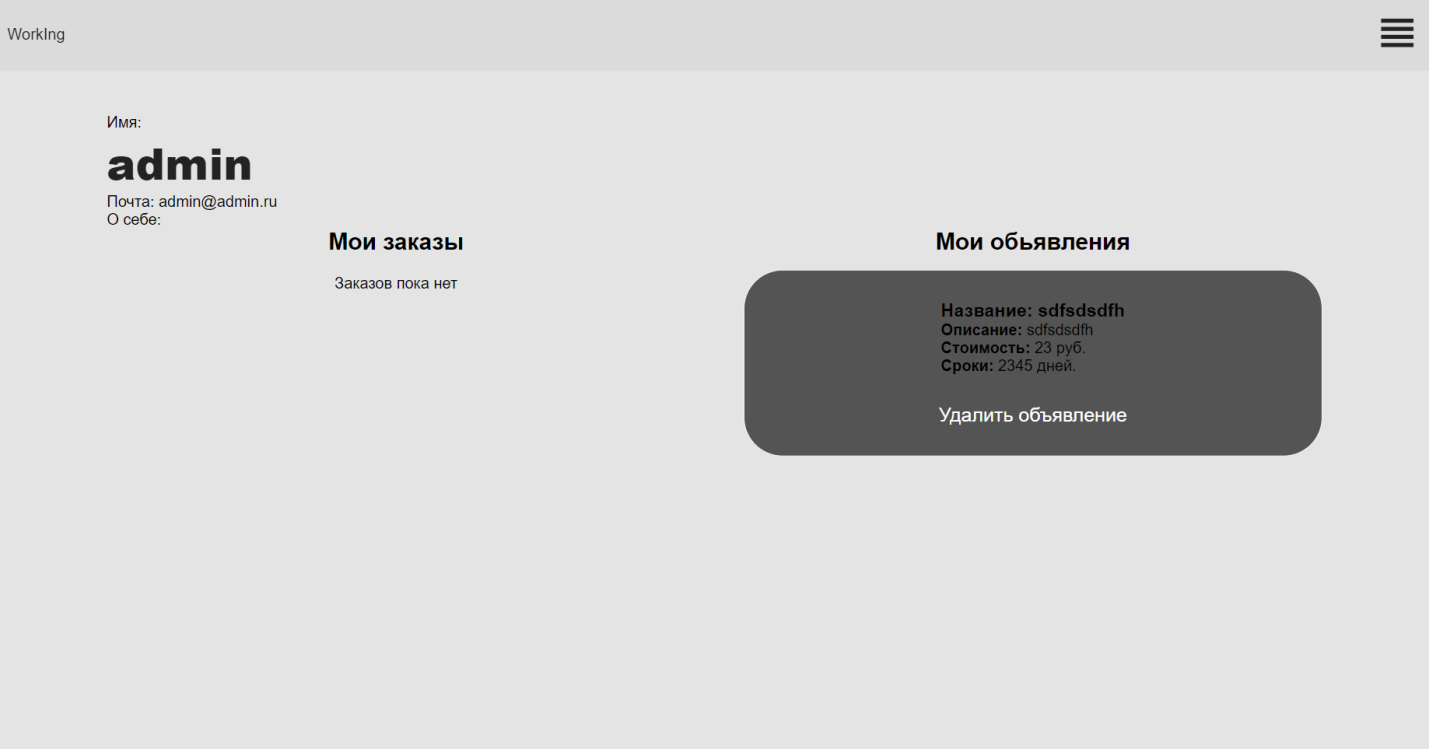


Рисунок 13.6 – личный кабинет исполнтеля

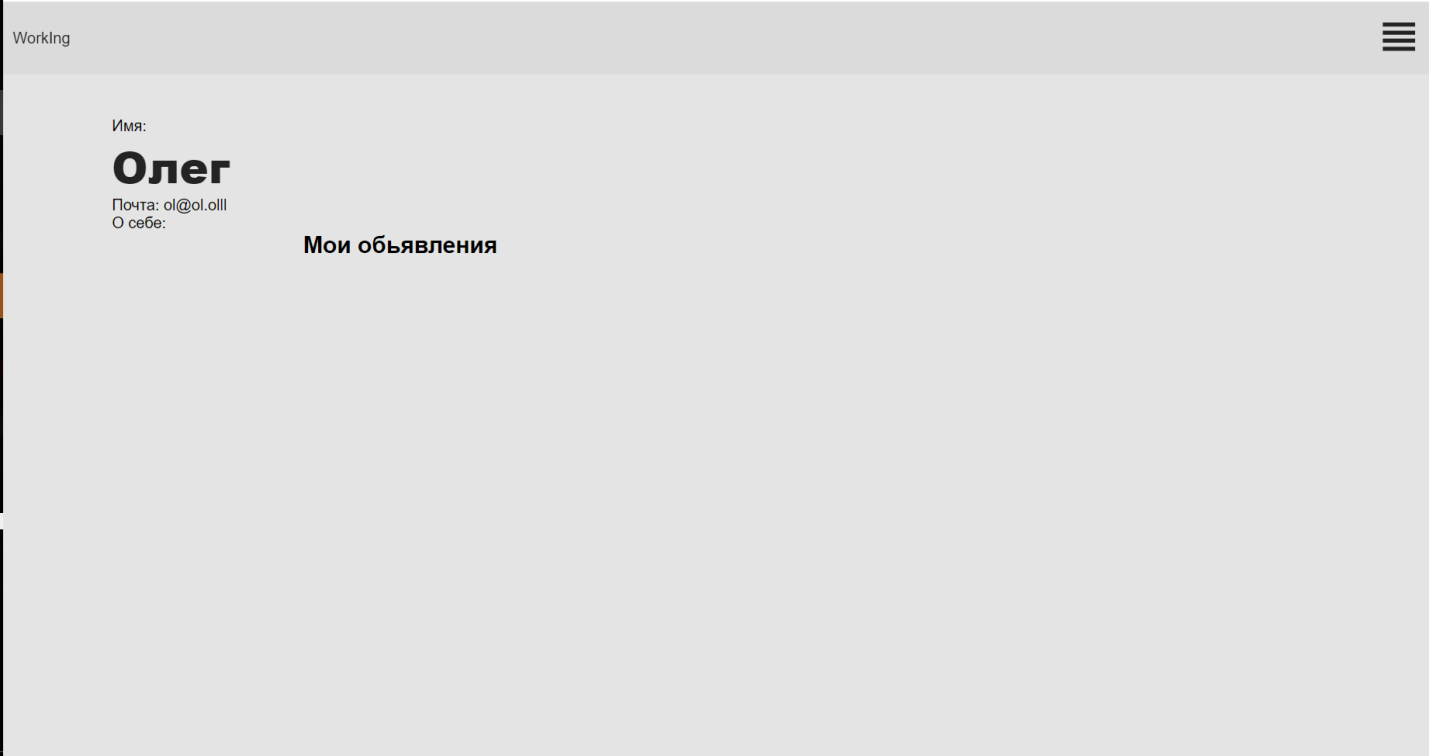


Рисунок 13.7 – личный кабинет клиента

У исполнителей есть возможность предложить свою кандидатуру на выполнение определённого заказа. Для этого нужно зайти на страницу объявления, выбрать объявление и нажать на кнопку «Подать кандидатуру». У клиентов такой возможности нет.

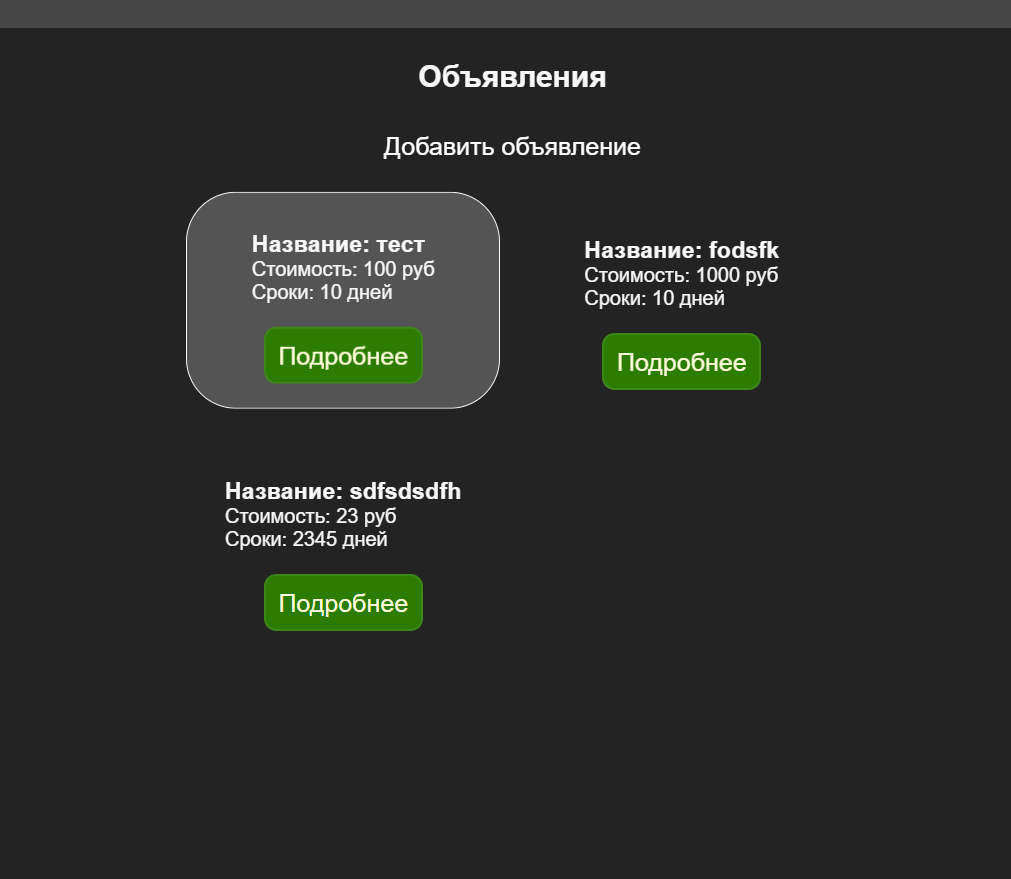


Рисунок 13.8 – станица «Объявления»

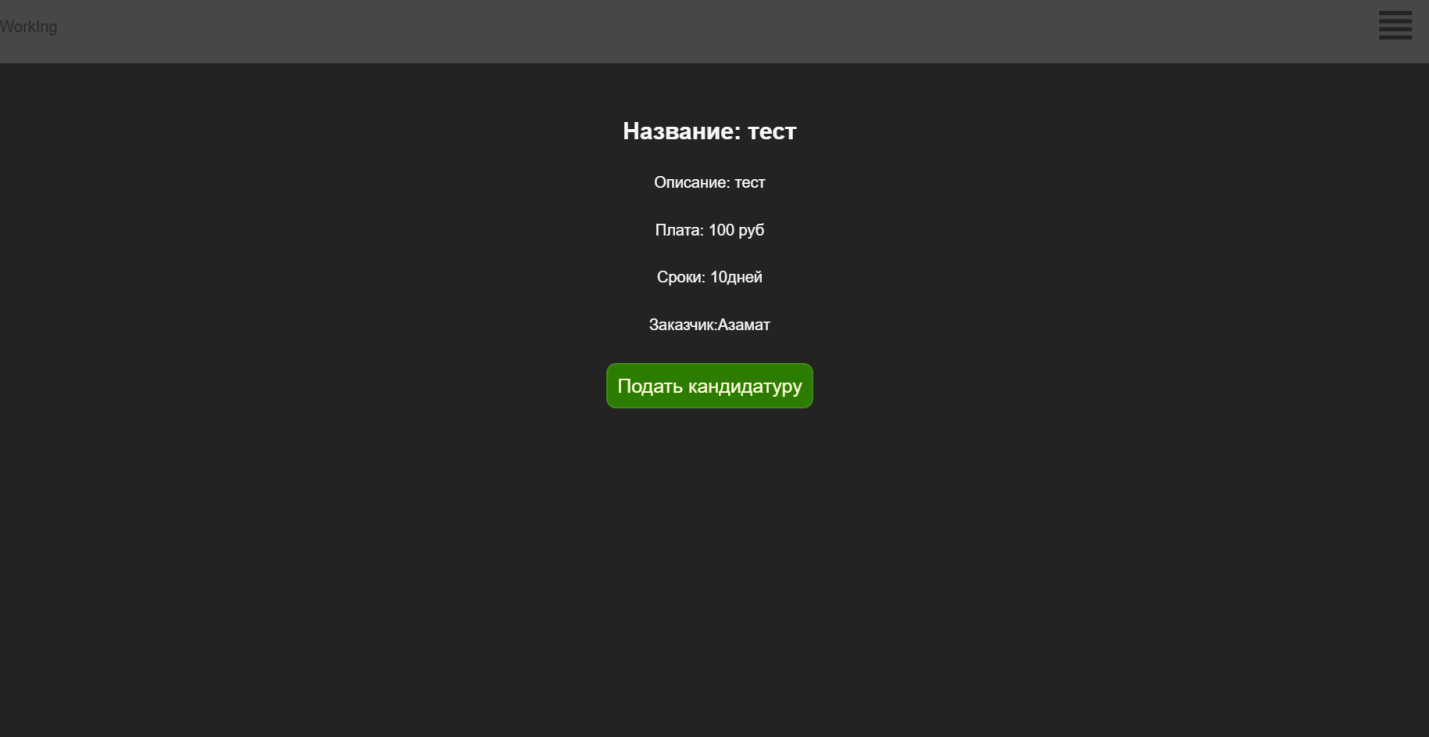


Рисунок 13.9 – детальная страница объявления

Так же у системы предоставляется возможность просмотра объявлений по их категориям.

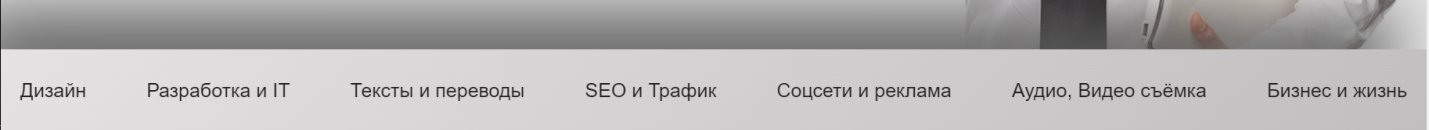


Рисунок 13.10 – категории на главной странице

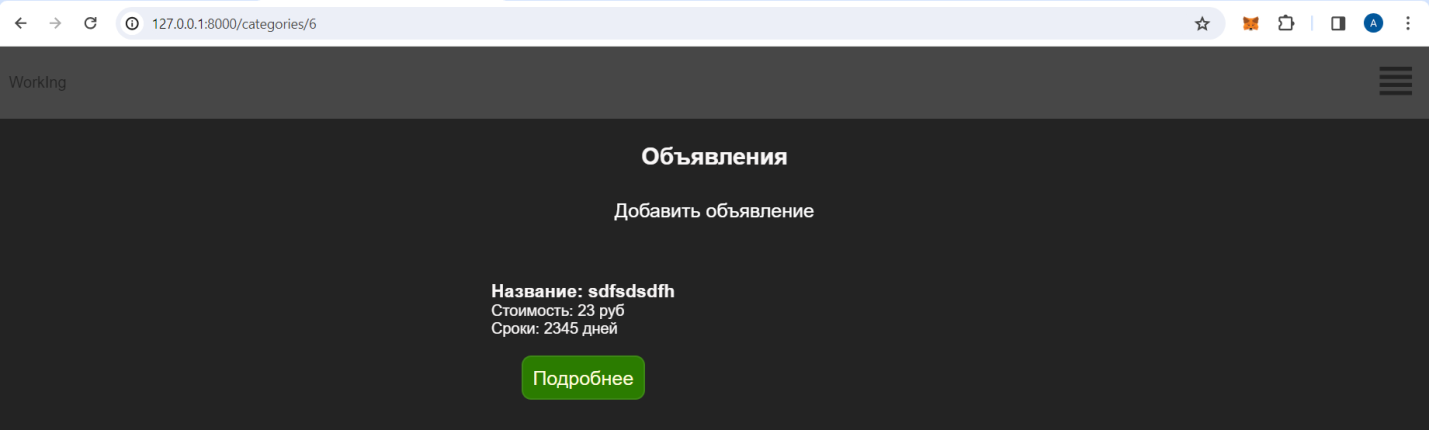


Рисунок 13.11 – страница с объявлениями по одной категории

Приложение А

UML диаграммы

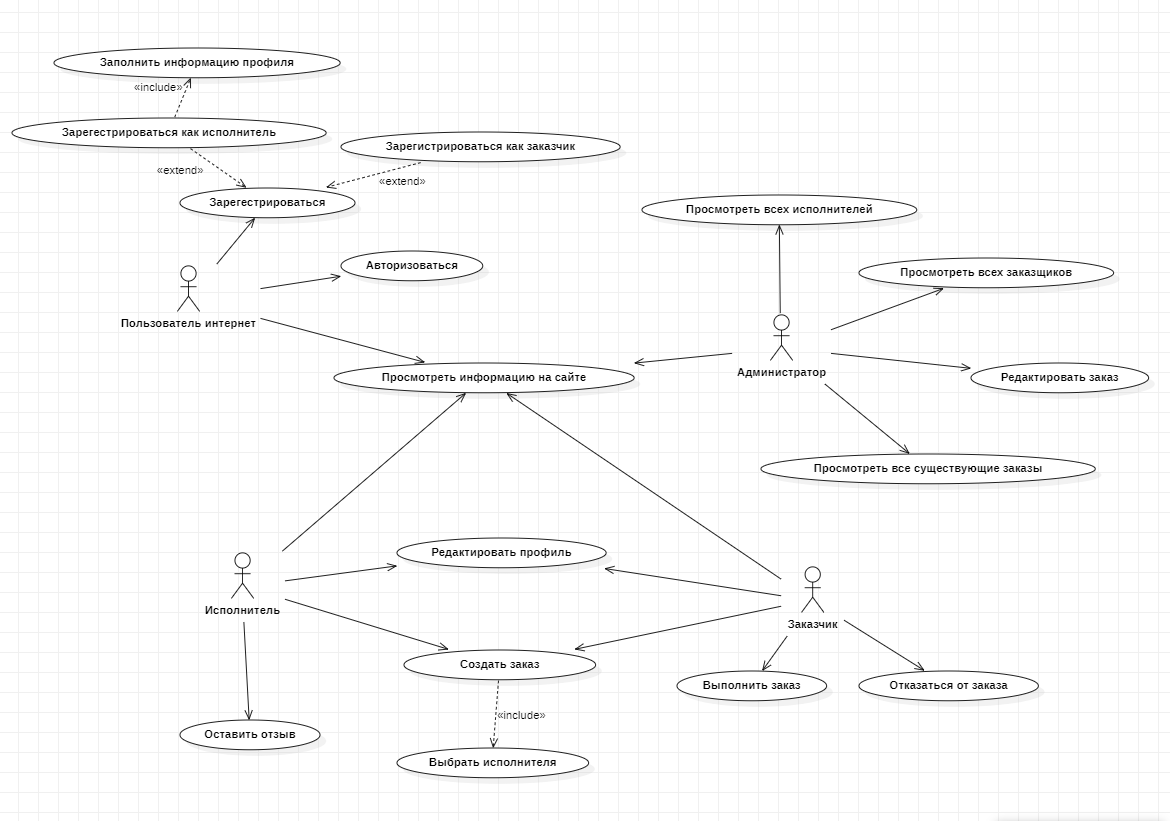


Рисунок А.1 – диаграмма прецедентов

Приложение Б

Логическая модель базы данных

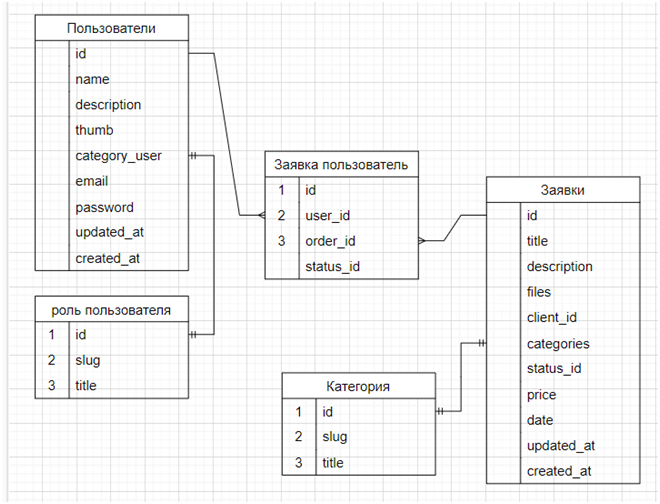


Рисунок Б.1 – даталогическая модель базы данных

Приложение В

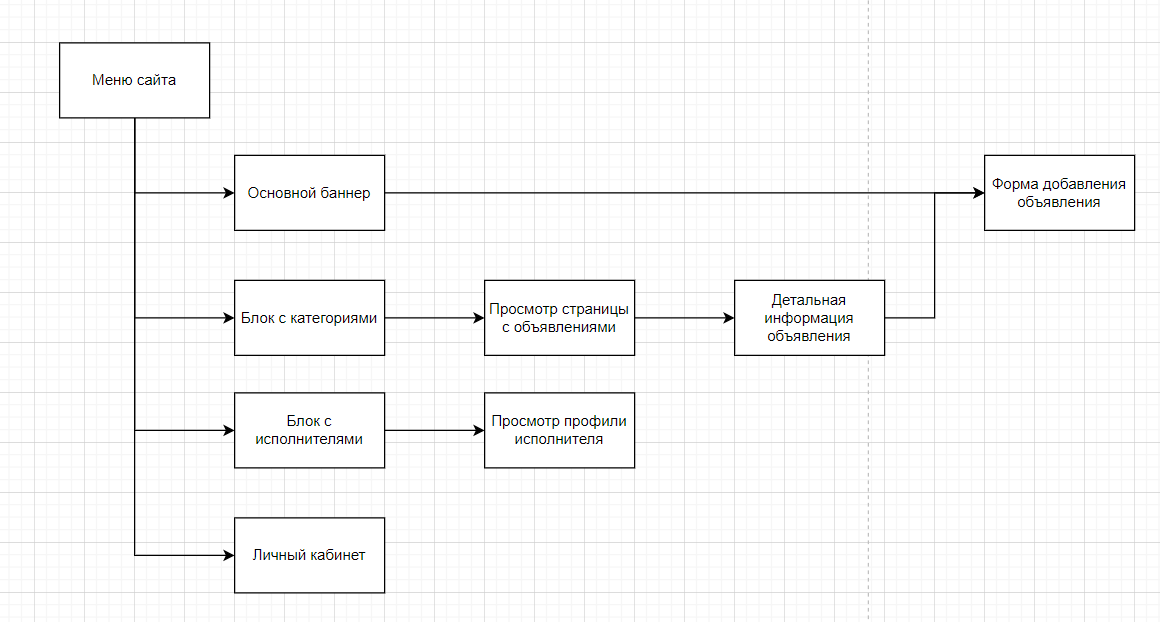


Рисунок В.1 – физическая модель сайта

СПИСОК ЛИТЕРАТУРЫ

1. ГОСТ 19.001–77. ЕСПД. Общие положения.
2. ГОСТ 19.003–80. ЕСПД. Схемы алгоритмов и программ. Обозначения условные графические. – Заменен на ГОСТ 19.701-90.
3. Википедия: офиц. сайт. URL: http://ru.wikipedia.org/wiki/
4. Язык запросов SQL: офиц. сайт. URL: https://sql-language.ru
5. Хабр: офиц. сайт. URL: https://habr.com/ru/all/
6. Htmlbook: офиц. сайт. URL: http://htmlbook.ru/
7. Html5 book: офиц. сайт. URL: http://html5book.ru/
8. Metanit: офиц. сайт. URL: https://metanit.com/
9. W3School: офиц. сайт. URL: https://www.w3schools.com/
10. itProger: офиц. сайт. URL: https://itproger.com/
11. Хекслет: офиц. сайт. URL: https://ru.hexlet.io/
12. Денвик: офиц. сайт. URL: https://denvic.ru/
13. JavaScript: офиц. сайт. URL: https://learn.javascript.ru/first-steps/
14. Laravel: офиц. сайт. URL: https://laravel.com/
15. Bootstrap Шпаргалка: офиц. сайт. URL: https://bootstrap-4.ru/articles/cheatsheet5/