УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ

«МИНСКИЙ ГОСУДАРСТВЕННЫЙ КОЛЛЕДЖ ЦИФРОВЫХ ТЕХНОЛОГИЙ»

Специальность: 2-40 01 01

«Программное обеспечение информационных технологий»

ДОПУЩЕН К ЗАЩИТЕ

Заведующий отделением

\_\_\_\_\_\_\_\_\_\_ Е.В. Макаренко

«\_\_\_»\_\_\_\_\_\_\_\_\_\_2025г

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Разработка веб-приложения для РЕСТОРАНА «HOKAIDO»

МГКЦТ ДП63ТП.022109.081 ПЗ

Автор проекта (А.П. Сушицкая )

Руководитель проекта (А.В. Варицкая )

Технический контроль (Е.А. Левонюк )

Нормоконтроль (А.В. Варицкая )

Консультант по экономической части (Е.С. Витебская )

Консультант по вопросам охраны труда (Н.А. Немцева )

Рецензент ( )

Допустить дипломный проект к защите в Государственной экзаменационной комиссии

Председатель цикловой комиссии (А.А. Шавейко )

2025

РЕФЕРАТ

Пояснительная записка 47 с., 7 р., 11 рис., 6 табл., 16 источников, 2 прил.

ВЕБ-ПРИЛОЖЕНИЕ, РЕСТОРАН, СТАЖЁРЫ, РАСПИСАНИЕ, ОБУЧЕНИЕ, МЕНЮ, АДМИНИСТРАТОР, БАЗА ДАННЫХ, ЭКОНОМИКА, TS, POSTGRESQL, РАСЧЁТ, ОХРАНА ТРУДА, ЭНЕРГОСБЕРЕЖЕНИЕ.

Объектом исследования является процесс организации стажировки и обучения новых сотрудников в ресторане.

Целью проекта является разработка веб-приложения, позволяющего стажёрам получать доступ к расписанию занятий, обучающим материалам, меню, а также взаимодействовать с наставниками и администрацией ресторана.

Программа разрабатывалась с использованием среды разработки PyCharm 2024, языка программирования Python и веб-фреймворка Django.

Цель работы – создание веб-приложения для автоматизации и систематизации процесса стажировки. Для незарегистрированных пользователей реализована возможность ознакомиться с общей информацией о ресторане и его принципах. Зарегистрированные пользователи (стажёры) получают доступ к личному кабинету, в котором они могут просматривать своё расписание, читать обучающие материалы, выполнять задания и отслеживать прогресс. Для администратора предусмотрены функции управления пользователями, добавления новых обучающих курсов, составления расписаний и отслеживания успеваемости стажёров.

Результатом работы стало веб-приложение «HOKAIDO», которое может использоваться персоналом ресторана для организации процесса стажировки и повышения эффективности обучения.

Основные эксплуатационные показатели: простота использования, удобный и интуитивно понятный интерфейс.

Степень внедрения – приложение используется стажёрами и администрацией ресторана для управления учебным процессом.

Эффективность программного средства: позволяет централизованно управлять процессом обучения, экономит ресурсы на проведение инструктажей, даёт возможность дистанционного контроля знаний.

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

Содержание

Изм.

Лист

№ документа

Подпись

Дата

Лист

4

МГКЦТ ДП63ТП.022109.081 ПЗ

Разраб.

Сушицкая А.П.

Провер.

Варицкая А.В.

Реценз.

Н. контр.

Варицкая А.В.

Утв.

*Шавейко А.А.*

*Разработка веб-приложения для ресторана «HOKAIDO»*

*Пояснительная записка*

Лит.

Листов

47

УО «МГКЦТ»

[ВВЕДЕНИЕ 6](#_Toc199843939)

[1 Подготовительный этап при создании веб-приложения 8](#_Toc199843940)

[1.1 Цели создания веб-приложения 8](#_Toc199843941)

[1.2 Проектирование содержимого веб-приложения 8](#_Toc199843942)

[1.3 Анализ аналогов 9](#_Toc199843943)

[2 Реализация 10](#_Toc199843944)

[2.1 Требование к веб-приложению и программному обеспечению 10](#_Toc199843945)

[2.2 Структура веб-приложения 11](#_Toc199843946)

[2.3 Дизайн веб-приложения 13](#_Toc199843947)

[2.4 Создание макета веб-приложения 13](#_Toc199843948)

[2.5 Программно-технические средства, необходимые для разработки веб-приложения 15](#_Toc199843949)

[2.6 Описание используемых библиотек и элементов управления 15](#_Toc199843950)

[2.7 Описание используемых функций и процедур 16](#_Toc199843951)

[2.8 Организация данных 18](#_Toc199843952)

[3 Методика испытаний 20](#_Toc199843953)

[3.1 Технические требования 20](#_Toc199843954)

[3.2 Функциональное тестирование 20](#_Toc199843955)

[4 Применение 23](#_Toc199843956)

[4.1 Назначение веб-приложения 23](#_Toc199843957)

[4.2 Программно-аппаратное обеспечение сервера и клиента 23](#_Toc199843958)

[5 Технико-экономическое обоснование дипломного проекта 25](#_Toc199843959)

[5.1 Краткая характеристика программного средства 25](#_Toc199843960)

[5.2 Расчет общей трудоемкости разработки программного обеспечения и трудоемкости отдельных стадий разработки 25](#_Toc199843961)

[5.3 Расчет сметы затрат на разработку программного обеспечения 27](#_Toc199843962)

[5.3.1 Затраты на основную заработную плату разработчика 27](#_Toc199843963)

[5.3.2 Затраты на дополнительную заработную плату разработчика 28](#_Toc199843964)

[5.3.3 Отчисления на социальные нужды 28](#_Toc199843965)

[5.3.4 Отчисления в Белгосстрах 29](#_Toc199843966)

[5.3.5 Прочие затраты 29](#_Toc199843967)

[5.4 Расчет экономического эффекта (прибыли) у разработчика программного обеспечения при реализации его заказчику 30](#_Toc199843968)

[6.1 Безопасная организация рабочего места программиста при разработке программного обеспечения 33](#_Toc199843969)

[6.1.1 Основные термины и определения 33](#_Toc199843970)

[7 Энерго- и ресурсосбережение 37](#_Toc199843973)

[ЗАКЛЮЧЕНИЕ 41](#_Toc199843974)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 42](#_Toc199843975)

Изм.

Лист

№ докум.

Подпись

Дата

Лист

5

МГКЦТ ДП63ТП.022109.081 ПЗ

[ПРИЛОЖЕНИЕ А 43](#_Toc199843976)

[ПРИЛОЖЕНИЕ Б 45](#_Toc199843977)

ВВЕДЕНИЕ

В последние годы наблюдается стремительное развитие информационных технологий, которые всё активнее внедряются в различные сферы человеческой деятельности – от промышленности до образования и сервиса. Одной из таких сфер является ресторанный бизнес, где цифровизация позволяет автоматизировать процессы обслуживания клиентов, управления персоналом и, что особенно важно, обучения новых сотрудников. [1]

В условиях высокой текучести кадров и необходимости поддержания высокого качества сервиса, особую актуальность приобретает вопрос эффективной организации стажировки. От того, насколько качественно и системно проходит обучение стажёров, зависит репутация заведения и удовлетворённость клиентов. Поэтому тема создания специализированных программных решений для поддержки стажировки в ресторане представляется особенно значимой. [2]

На практике обучение часто осуществляется вручную, с применением устаревших методик, без централизованного контроля знаний и гибкого планирования. Это может привести к несогласованности в действиях сотрудников, потере информации и снижению качества подготовки. Также наблюдается нехватка удобных цифровых инструментов, которые бы позволяли администратору отслеживать прогресс каждого стажёра, предоставлять актуальные материалы и формировать удобное расписание. [3]

В связи с этим темой проекта является разработка веб-приложения «HOKAIDO», предназначенного для автоматизации процесса стажировки новых сотрудников в ресторане.

Для достижения поставленной цели решались следующие задачи:

* реализация системы ролей: стажёр, наставник, администратор;
* разработка личного кабинета стажёра с расписанием, материалами, заданиями и обратной связью;
* организация панели администратора с возможностью добавления, редактирования и удаления курсов, расписаний и пользователей;
* обеспечение фильтрации данных (по сменам, наставникам, дате);
* создание базы готовых обучающих программ и модулей;
* разработка простого, интуитивного интерфейса и адаптивного дизайна.

Таким образом, реализация проекта HOKAIDO предоставляет удобный и функциональный инструмент для стажировки сотрудников в ресторане, повышая эффективность подготовки кадров и снижая нагрузку на наставников и менеджеров.

Пояснительная записка к дипломному проекту включает в себя ряд разделов:

* «Введение» – обзор и оценка современного состояния решаемой проблемы;
* «Подготовительный этап при создании веб-приложения» – содержит описание предметной области, анализ аналогов;
* «Реализация» – заключается в кодировании программного продукта, рассматриваются данные, процессы и инструменты разработки в комплексе;
* «Методика испытаний» – заключается в тестировании программного продукта;
* «Применение» – название и назначение веб-приложения;
* «Вопросы охраны труда при работе с компьютерами» – описывается безопасная организация рабочего места программиста при разработке программного обеспечения;
* «Энерго- и ресурсосбережение» – информация о энергосбережении и ресурсосбережении;
* «Заключение» – краткая формулировка проблемы, пути решения проблемы, использованные методы и средства, возможность модификации;
* «Список использованных источников» – перечень источников, из которых бралась информация для реализации поставленной задачи;
* «Приложения» – схемы, таблицы, рисунки, не вошедшие в основную часть программы.

# 1 Подготовительный этап при создании веб-приложения

# 1.1 Цели создания веб-приложения

Разработка веб-приложения «HOKAIDO» направлена на автоматизацию и упрощение процесса стажировки сотрудников в ресторане. Приложение предназначено для стажёров, администраторов и наставников, предоставляя каждому доступ к функционалу, соответствующему их роли. Пользователи могут просматривать расписание, обучающие материалы, выполнять задания и отслеживать собственный прогресс, а администрация – управлять учебным процессом и контролировать успеваемость.

Создание такого приложения даёт ряд важных преимуществ:

* возможность централизованного обучения сотрудников через интернет;
* повышение вовлечённости стажёров за счёт понятного и удобного интерфейса;
* снижение нагрузки на наставников и администраторов;
* повышение качества подготовки персонала и, как следствие, уровня сервиса в ресторане.

Потенциальной аудиторией разрабатываемого веб-приложения являются совершеннолетние граждане Республики Беларусь и гости страны.

# Проектирование содержимого веб-приложения

Веб-приложение «HOKAIDO» будет включать в себя следующий функционал:

* отображение расписания смен и обучающих мероприятий для стажёров;
* предоставление доступа к обучающим материалам (видео, текстовые инструкции, тесты);
* отслеживание прогресса стажёров и результатов тестирования;
* возможность обратной связи с наставником (вопросы, комментарии, отчёты);
* публикация новостей и обновлений, связанных с работой ресторана и обучением персонала;
* фильтрация расписания и материалов по дате, наставнику, типу смены;
* система уведомлений (о новых заданиях, изменениях в расписании и т.д.);
* административная панель – интерфейс для администратора с возможностью создания, редактирования и удаления расписаний, обучающих модулей, пользователей и новостных публикаций.

# 1.3 Анализ аналогов

На сегодняшний день аналогов разрабатываемого веб-приложения «HOKAIDO» в открытом доступе практически не существует. Большинство ресторанов используют разрозненные подходы к обучению новых сотрудников: от печатных инструкций до устных наставлений и локальных чат-групп. Подобные методы зачастую не позволяют организовать полноценный и структурированный процесс стажировки, не обеспечивают сохранности обучающих материалов, а также не предоставляют возможности отслеживать прогресс стажёра.

«HOKAIDO» выгодно отличается на фоне таких подходов, предлагая централизованную платформу для автоматизации всего процесса стажировки. В отличие от типовых корпоративных LMS-систем, которые ориентированы на крупные компании и требуют сложной настройки, «HOKAIDO» адаптировано под нужды небольших и средних ресторанов, имеет понятный интерфейс и ориентировано на простого пользователя.

Проект объединяет в себе такие функции, как:

* доступ к расписанию и материалам в личном кабинете стажёра,
* обратная связь с наставником,
* встроенная система уведомлений,
* административная панель для управления всеми аспектами обучения.

Таким образом, веб-приложение «HOKAIDO» является уникальным решением, позволяющим организовать качественную, гибкую и эффективную систему стажировки в ресторанном бизнесе.

# 2 Реализация

# 2.1 Требование к веб-приложению и программному обеспечению

Разрабатываемый продукт – это веб-приложение для организации и проведения стажировки в ресторане. Основная цель – обеспечить централизованный, прозрачный и эффективный процесс обучения новых сотрудников с доступом к необходимым материалам, расписаниям и инструментам коммуникации. Приложение направлено на повышение качества подготовки стажёров и оптимизацию работы наставников и администраторов.

Исходя из этой цели, приложение должно соответствовать следующим требованиям:

* предоставление стажёру доступа к расписанию, материалам и заданиям;
* возможность прохождения тестирования и отслеживания результатов;
* система уведомлений и обратной связи с наставником;
* административная панель с возможностью управления пользователями, материалами, расписанием и новостями;
* современный, интуитивно понятный интерфейс, адаптированный под разные устройства.

Первоначальная структура веб-приложения:

а) основной пользовательский интерфейс:

1. главная страница;
2. страница обучающих материалов;
3. расписание стажировок;
4. новости ресторана;
5. личный кабинет стажёра;
6. страница обратной связи;
7. страница «О нас»;
8. страница «Справка».

б) административная панель:

1. управление пользователями (стажёры, сотрудники, администраторы);
2. управление обучающими модулями и тестами;
3. редактирование расписания и новостей;
4. просмотр статистики успеваемости.

Требования к графическому дизайну:

Стиль: современный, адаптированный под цифровую среду обучения.

Основные цвета: оранжевый (#FFAA00), белый (#FFFFFF), синий (#06607B).

Цвет текста: тёмно-синий или белый (в зависимости от фона).

Логотип: фирменный логотип проекта.

Графические элементы: иконки, изображения и иллюстрации на всех ключевых страницах.

Фон: белый или синий, с чередованием тёмных и светлых блоков для удобства восприятия.

Шрифт: Montserrat.

Макет: адаптивная верстка с автоматическим масштабированием, минимальная ширина рабочего поля – 700 пикселей, минимальная высота – 600 пикселей.

На всех страницах должно быть шапка сайта с логотипом, навигацией и ссылками на соцсети и подвал сайта с дублирующими навигационными ссылками, контактной информацией, страницами «О нас» и «Справка».

# 2.2 Структура веб-приложения

Организационная структура страниц веб-приложения представлена на рисунке 2.1.

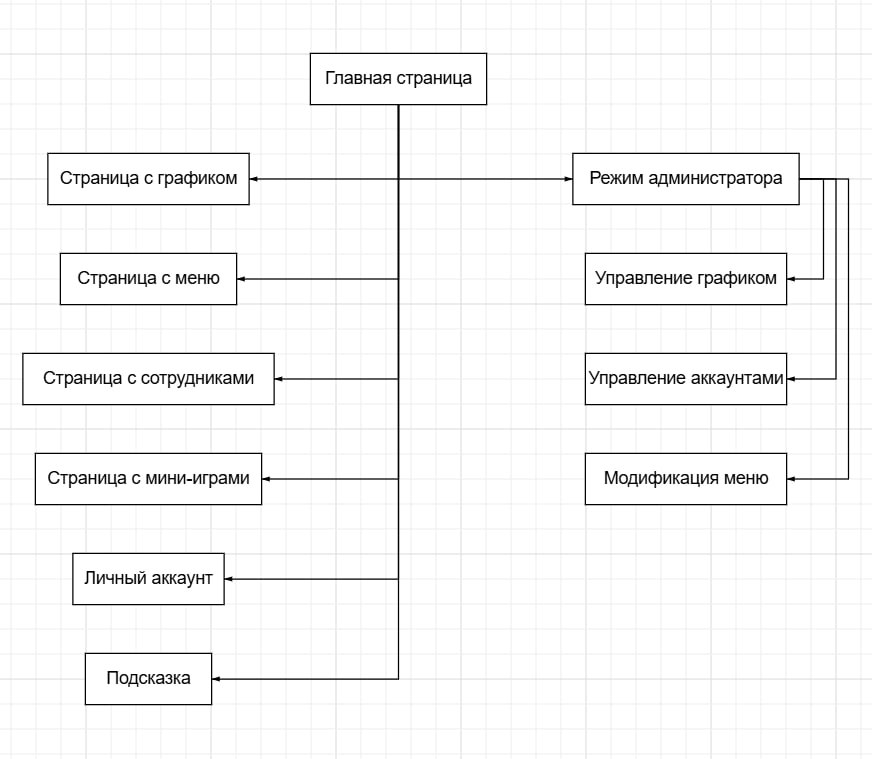


Рисунок 2.1 – Структура страниц веб-приложения

Физическая структура визуальной части приложения представлена на рисунке 2.2.

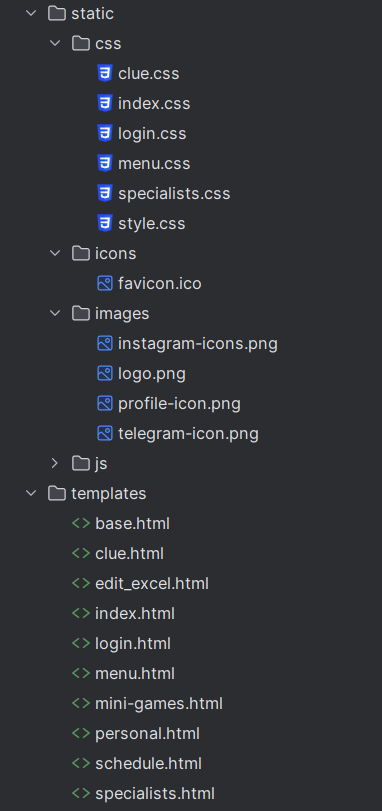


Рисунок 2.2 – Файловая структура визуальной части проекта

Физическая структура серверной части представлена на рисунке 2.3.

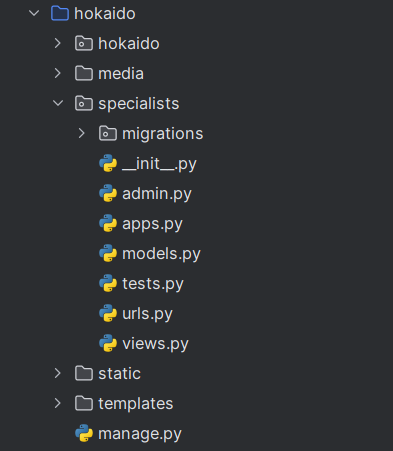


Рисунок 2.3 – Файловая структура сервера приложения

# 2.3 Дизайн веб-приложения

Макет и дизайн определяют, как будет представлена информация в вашем веб-приложении. Для разработки приложения «HOKAIDO» был использован Django, который позволяет создавать интерактивные веб-интерфейсы и легко интегрируется с популярными JavaScript-фреймворками, такими как Angular, React, Vue, а также с библиотекой стилей Bootstrap.

Для веб-приложения «HOKAIDO» также использована типичная модель построения модулей сайта, а именно базовый шаблон, разбитый на основную часть с контентом и две другие: футер, хедер. Остальные страницы (кроме страниц администратора) наследуются от базового шаблона и расширяют его контентом. На главной странице кроме футера и хедера находиться информационный слайдер, данные о сотрудниках и новости.

Прототип главной страницы веб-приложения представлен на рисунке 2.4.

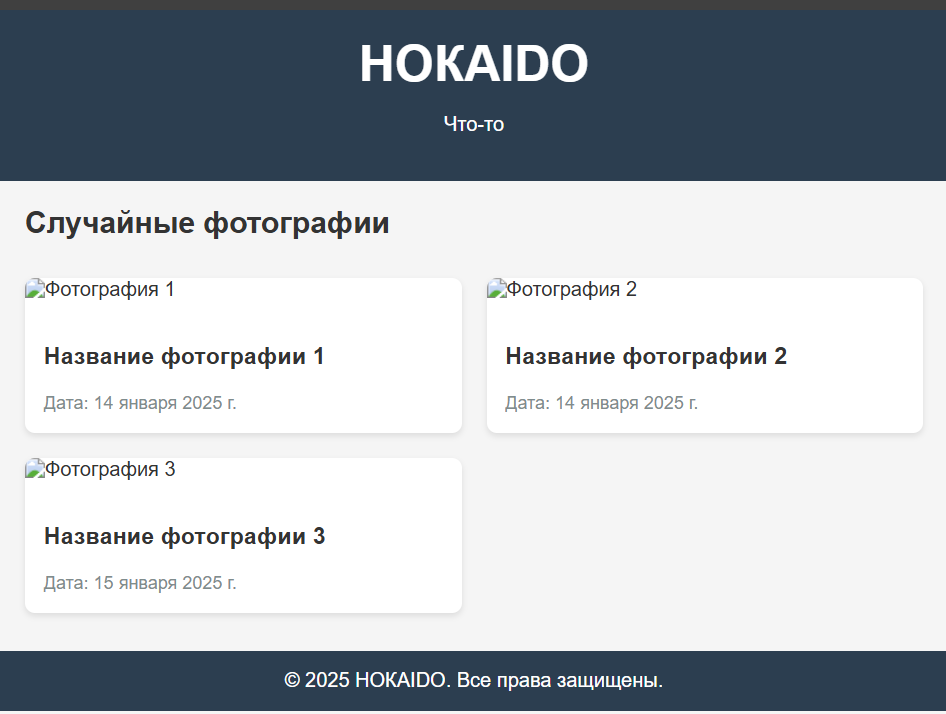


Рисунок 2.4 – Прототип главной страницы

# 2.4 Создание макета веб-приложения

Для создания макета веб-приложения был выбран инструмент Figma. Figma представляет собой мощный веб-приложение для дизайна интерфейсов, которое позволяет создавать и прототипировать различные типы дизайн-элементов, включая макеты сайтов. Он обладает широким набором функций, интуитивно понятным пользовательским интерфейсом и возможностью коллаборации в реальном времени, что делает его популярным выбором среди дизайнеров и команд разработки.

Ширина макета веб-приложения составляет 1280px. Фон макета #FFFFFF.

При создании основного блока веб-приложения были использованы инструменты: «Круг», «Прямоугольник» и «Направленная линия». Ширина основного блока составляет 960px. Блок готовых сборок содержит несколько собранных компьютеров, блок новостей включает в себя заголовки новостей, которые находятся на соответствующей странице. Основная цветовая схема: #FFFFFF.

При создании шапки и футера веб-приложения были использованы инструменты: «Круг», «Прямоугольник» и «Изображение». Ширина подвала составляет 960px. Шапка веб-приложения содержит ссылку на главную страницу, обратную связь и о нас. Основная цветовая гамма - #FFFFFF.

Для создания дизайна веб-предложения был использован фреймворк NestJS, а для стилизации и компоновки блоков веб-приложения был использован CSS-модульный подход с применением свойства flex. Flex позволяет легко управлять расположением и выравниванием элементов на странице. Для блоков веб-приложения были применены личные свойства, например flex-basis, которое изменяет размер элемента внутри контейнера. Это позволяет гибко управлять размерами, порядком и выравниванием блоков. Также было использовано множество дополнительных классов для стилизации различных элементов, таких как заголовки, ссылки кнопки.

Макет главной страницы веб-приложения представлен на рисунке 2.5.



Рисунок 2.5 – Макет главной страницы

# 2.5 Программно-технические средства, необходимые для разработки веб-приложения

Данный дипломный проект был написан с использованием Django и PostgreSQL.

Django – это высокоуровневый веб-фреймворк на языке Python, который способствует быстрому развитию и чистому, прагматичному дизайну. Он предлагает мощные инструменты для создания современных, высокопроизводительных веб-решений, включая веб-сайты, API и административные панели.

Основные характеристики Django:

* кроссплатформенность: Django работает на различных операционных системах, включая Windows, macOS и Linux, что делает его доступным для разработчиков на любой платформе;
* встроенная модульность: Django предлагает ряд встроенных модулей, таких как ORM для работы с базами данных, система аутентификации и админ-панель, что ускоряет разработку и упрощает структуру проекта;
* поддержка различных типов приложений: Django позволяет создавать как традиционные веб-приложения с использованием шаблонов и форм, так и RESTful API с помощью Django REST Framework;
* современные возможности: Django поддерживает асинхронное программирование и может эффективно обрабатывать множество запросов, улучшая производительность и отзывчивость приложений;
* интеграция с современными инструментами: Django легко интегрируется с библиотеками и инструментами, такими как Celery для фоновых задач и Pytest для тестирования, а также поддерживает расширение функциональности через плагины и пакеты;
* безопасность: Django включает встроенные механизмы защиты от распространенных уязвимостей, таких как XSS и CSRF, а также поддерживает сложные системы аутентификации и авторизации;
* развиваемая экосистема: Django имеет активное сообщество, множество пакетов и обширную документацию, что облегчает добавление новых функций и решение возникающих вопросов;
* сообщество и документация: Django обладает мощным сообществом разработчиков и обширной документацией, что делает его идеальным выбором для обучения и реализации сложных проектов. [7]

# 2.6 Описание используемых библиотек и элементов управления

В качестве СУБД для данного проекта был выбран PostgreSQL – мощная реляционная база данных с открытым исходным кодом, известная своей надежностью, производительностью и богатым набором функций. PostgreSQL обладает следующими ключевыми преимуществами:

* масштабируемость и производительность: PostgreSQL поддерживает сложные запросы и большие объемы данных, что делает его идеальным выбором для веб-приложений с высокими требованиями к производительности;
* совместимость с различными платформами: PostgreSQL доступен на многих операционных системах, включая Windows, macOS и Linux, что позволяет использовать его в различных средах;
* поддержка расширений: PostgreSQL имеет широкий спектр расширений, таких как PostGIS для работы с географическими данными, что позволяет легко наращивать функциональность базы данных; [8]
* надежность и устойчивость: благодаря механизму транзакций ACID, PostgreSQL обеспечивает сохранение целостности данных даже при сбоях;
* расширенная поддержка типов данных: помимо стандартных типов, PostgreSQL поддерживает JSON, XML, и даже собственные пользовательские типы, что дает разработчикам гибкость в хранении и обработке данных; [9]
* открытый исходный код: PostgreSQL является проектом с открытым исходным кодом, что обеспечивает прозрачность и возможность модификации.

PostgreSQL широко используется в веб-приложениях, корпоративных системах и для разработки сложных аналитических решений, а также в таких проектах, как Instagram, Skype и Spotify. В целом, PostgreSQL является мощным инструментом для управления данными, обладающим высокой надежностью, масштабируемостью и производительностью в условиях больших объемов данных и высоких нагрузок.

Пользователи могут легко взаимодействовать с базой данных, а разработчики имеют доступ ко всем функциям и возможностям PostgreSQL, что позволяет эффективно разрабатывать приложения. [10]

Для создания веб-приложения использовались следующие ключевые библиотеки и фреймворки:

* Django: основной фреймворк для разработки веб-приложения, обеспечивающий быстрый старт, безопасность и удобные инструменты для работы с базами данных; [11]
* Django REST Framework: использовался для создания API, обеспечивая доступ к данным и их обработку в формате JSON; [12]

# 2.7 Описание используемых функций и процедур

Для реализации поставленных задач было использовано большое количество функций. Некоторые из них:

* функция для загрузки главной страницы:

def index(request):

specialists = Account.objects.filter(position\_\_isnull=False) # Фильтруем по наличию позиции, если нужно

# Получаем все фотографии

photogallery = PhotoGallery.objects.all()

# Выбираем 3 случайные фотографии

random\_photos = random.sample(list(photogallery), 3) if len(photogallery) >= 3 else photogallery

return render(request, 'index.html', {'specialists': specialists, 'random\_photos': random\_photos})

* доступы к страницам:

urlpatterns = [

path('main/', views.index, name='index'),

path('', auth\_views.LoginView.as\_view(template\_name='login.html'), name='login'),

path('schedule/', views.schedule\_view, name='schedule'),

path('menu/', views.menu\_view, name='menu'),

path('mini\_games/', views.mini\_games\_view, name='mini\_games'),

path('specialists/', views.specialists\_view, name='specialists'),

path('clue/', views.clue\_view, name='clue'),

path('personal/', views.personal\_view, name='personal'),

] + static(settings.MEDIA\_URL, document\_root=settings.MEDIA\_ROOT)

* модель для меню:

class Menu(models.Model):

type\_of\_dish = models.ForeignKey(

'TypeOfDish',

on\_delete=models.CASCADE,

verbose\_name="Тип блюда", # Исправлено на "Тип блюда" вместо "Должность"

related\_name="menus"

)

name = models.CharField(max\_length=200, unique=True, verbose\_name="Название")

photo = models.ImageField(upload\_to='menu/photos/', blank=True, null=True, verbose\_name="Фото блюда")

compound = models.TextField(verbose\_name="Состав")

class Meta:

verbose\_name = "Меню"

verbose\_name\_plural = "Меню"

def \_\_str\_\_(self):

return self.name

# 2.8 Организация данных

Организация данных представляет собой логическую и физическую структуру в контексте среды разработки. Основным объектом базы данных является таблица. При создании таблиц сначала определяется состав, имена, тип данных, ключи, затем создается база данных. База приложения содержит двенадцать таблиц, основные из них – это «Account», «Menu», «Position». Их подробное описание представлено в таблицах 2.1 – 2.4.

Таблица 2.1 – Класс «Account»

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Наименование (смысл отображаемой информации) | Тип поля | Примечание |
| Id | Идентификатор | Числовой | Уникальный |
| last\_name | Фамилия | Текстовый |  |
| first\_name | Имя | Текстовый |  |
| middle\_name | Отчество | Текстовый |  |
| date\_of\_birth | Дата рождения | Дата |  |
| photo | Фотография | Текстовый |  |
| position | Должность | Числовой | Вторичный ключ |
| phone | Номер телефона | Текстовый |  |
| email | Email | Текстовый | Уникальный |
| additional\_info | Дополнительная информация | Текстовый |  |

Таблица 2.2 – Класс «Menu»

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Наименование (смысл отображаемой информации) | Тип поля | Примечание |
| Id | Идентификатор | Числовой | Уникальный |
| type\_of\_dish | Тип блюда | Числовой | Вторичный ключ |
| name | Название блюда | Текстовый | Уникальный |
| photo | Фото блюда | Текстовый |  |
| compound | Состав блюда | Текстовый |  |

Таблица 2.3 – Класс «Position»

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Наименование (смысл отображаемой информации) | Тип поля | Примечание |
| Продолжение таблицы | 2.3 |  |  |
| Id | Идентификатор | Числовой | Уникальный |
| title | Название должности | Текстовый | Уникальный |
| description | Описание должности | Текстовый |  |

Схема базы данных веб-приложения представлена на рисунке 2.6.

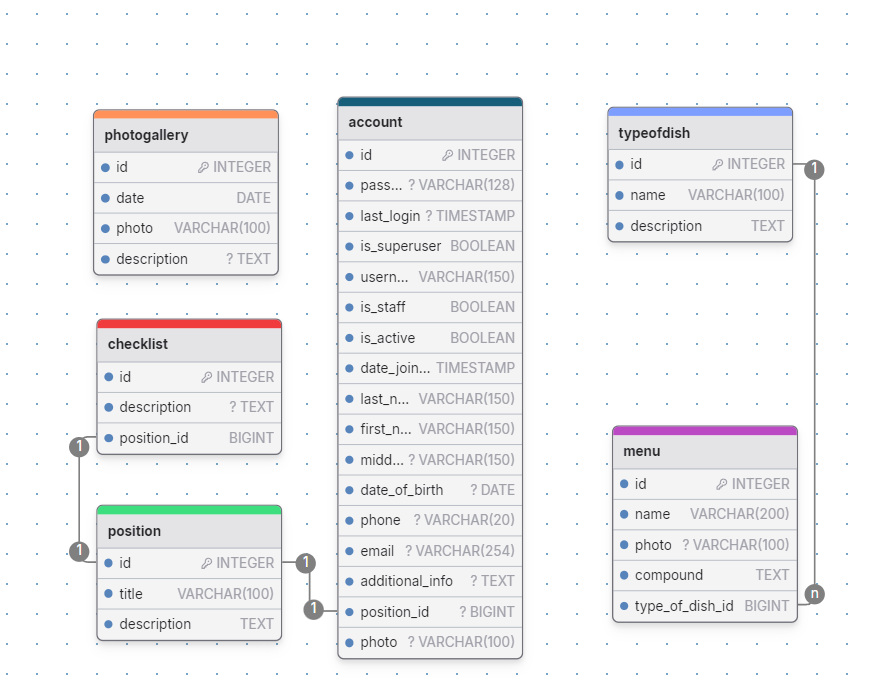


Рисунок 2.6 – Схема базы данных веб-приложения

# 3 Методика испытаний

# 3.1 Технические требования

Для полноценного функционирования разработанного веб-приложения, качественного выполнения всех поставленных задач, требуется ПК следующей конфигурации:

* процессор – Intel Core i3-8100;
* оперативная память – 8 Гбайта и более;
* дисковой накопитель – 500 Гбайт и более;
* операционная система Windows.

Для проведения тестирования совместимости веб-приложения с различными браузерами требуются браузеры: Opera (последних версий), Google Chrome (последних версий), Mozilla Firefox (последних версий).

# 3.2 Функциональное тестирование

Функциональное тестирование – это тестирование функций приложения на соответствие требованиям. Оценка производится в соответствии с ожидаемыми и полученными результатами. Результат функционального тестирования представлен далее.

Во время тестирования будут проверяться все функции, доступные в веб-приложении: просмотр данных, вход в аккаунт, запись обратной связи, решение тестов.

После чего, все собранные и найденные ошибки будут исправлены.

Главная страница веб-приложения представлена на рисунке 3.1.

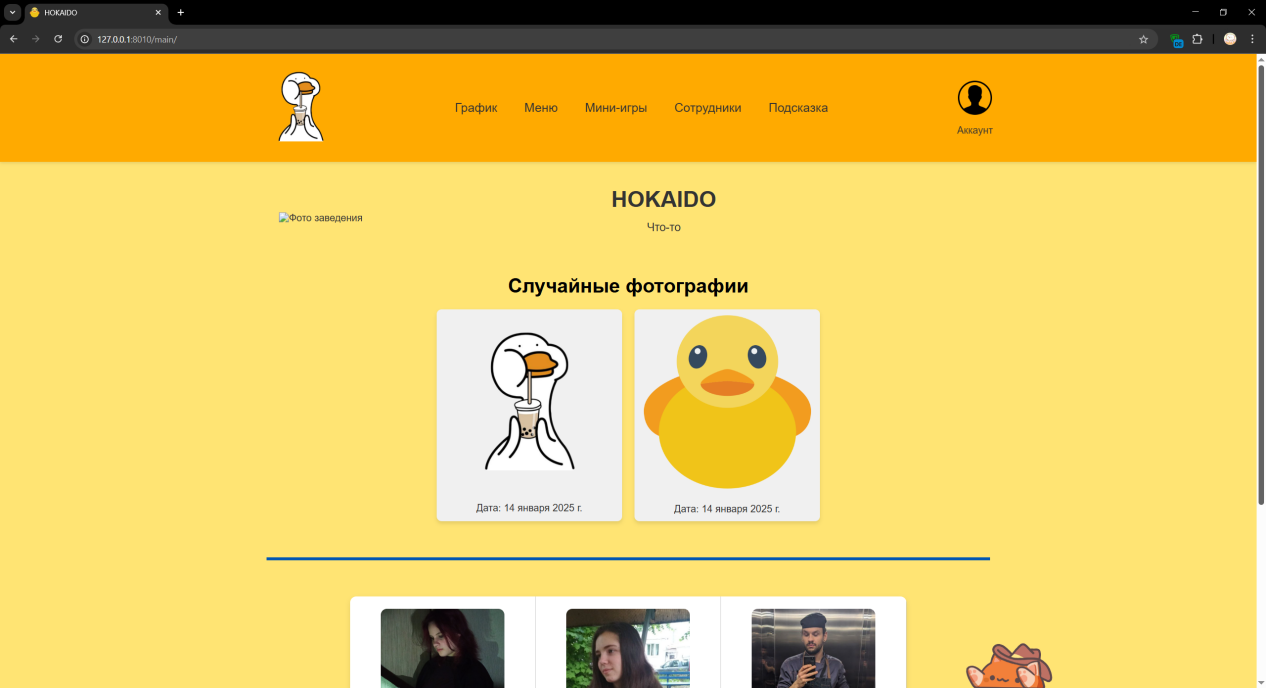


Рисунок 3.1 – Главная страница веб-приложения

В ходе тестирования веб-приложения была проведена серия проверок, направленных на оценку корректности загрузки и отображения различных компонентов пользовательского интерфейса.

Начальной проверкой стала возможность входа в аккаунт. Тестировщик открыл веб-приложение, после чего ввёл данные предоставленного ему аккаунта и успешно переместился на главную страницу. Результат предоставлен на рисунке 3.2.

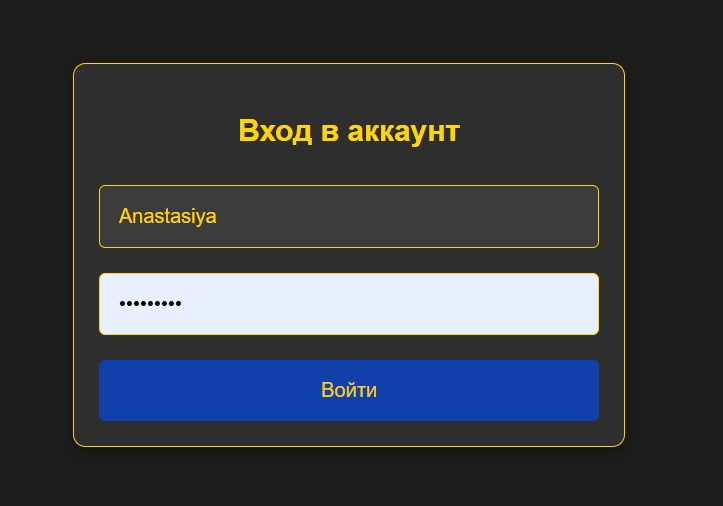


Рисунок 3.2 – Страница входа в аккаунт

Второй проверкой стало прохождение тестов. Тестировщик перешёл на страницу мини-игр, выбрал тест «Гунканы», выбрал тест «угадай блюдо», прошёл тест. В результате тест был успешно пройден, резултат прохождения выведен на экран, как и ожидалось. Данный результат показан на рисунках 3.2 - 3.5.

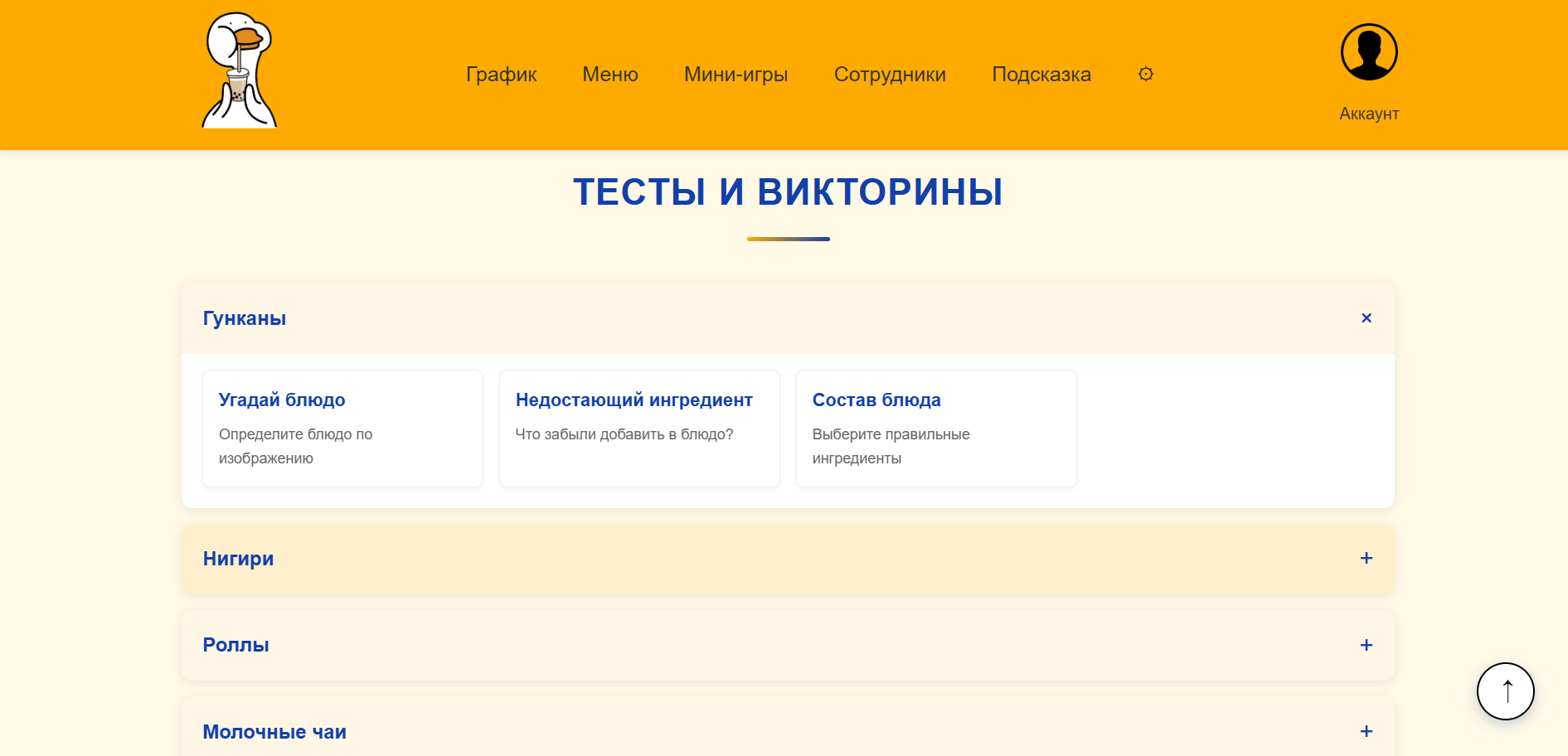


Рисунок 3.3 – Страница с мини-играми

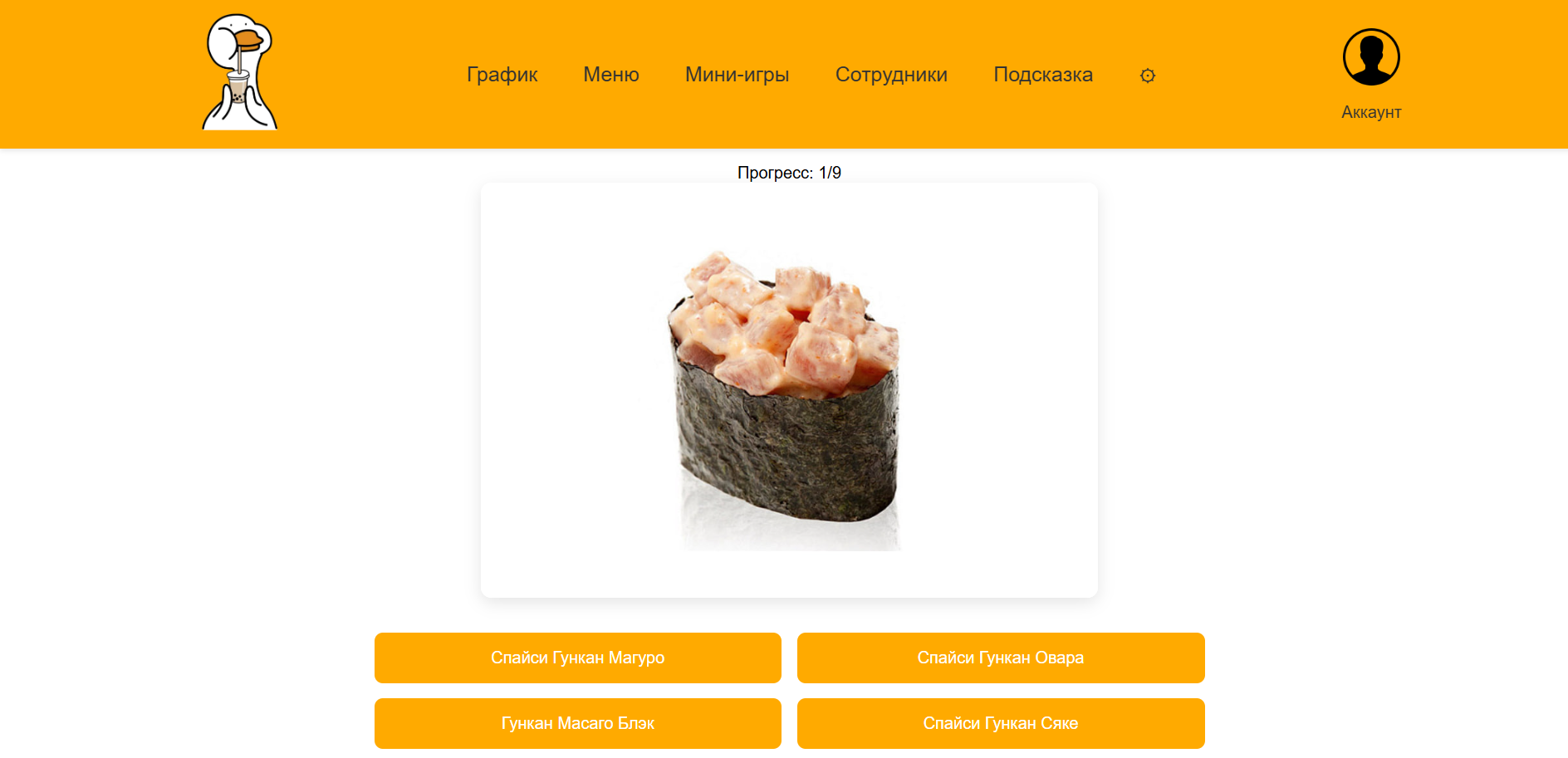
****

Рисунок 3.4 – Страница мини-игры «угадай блюдо»

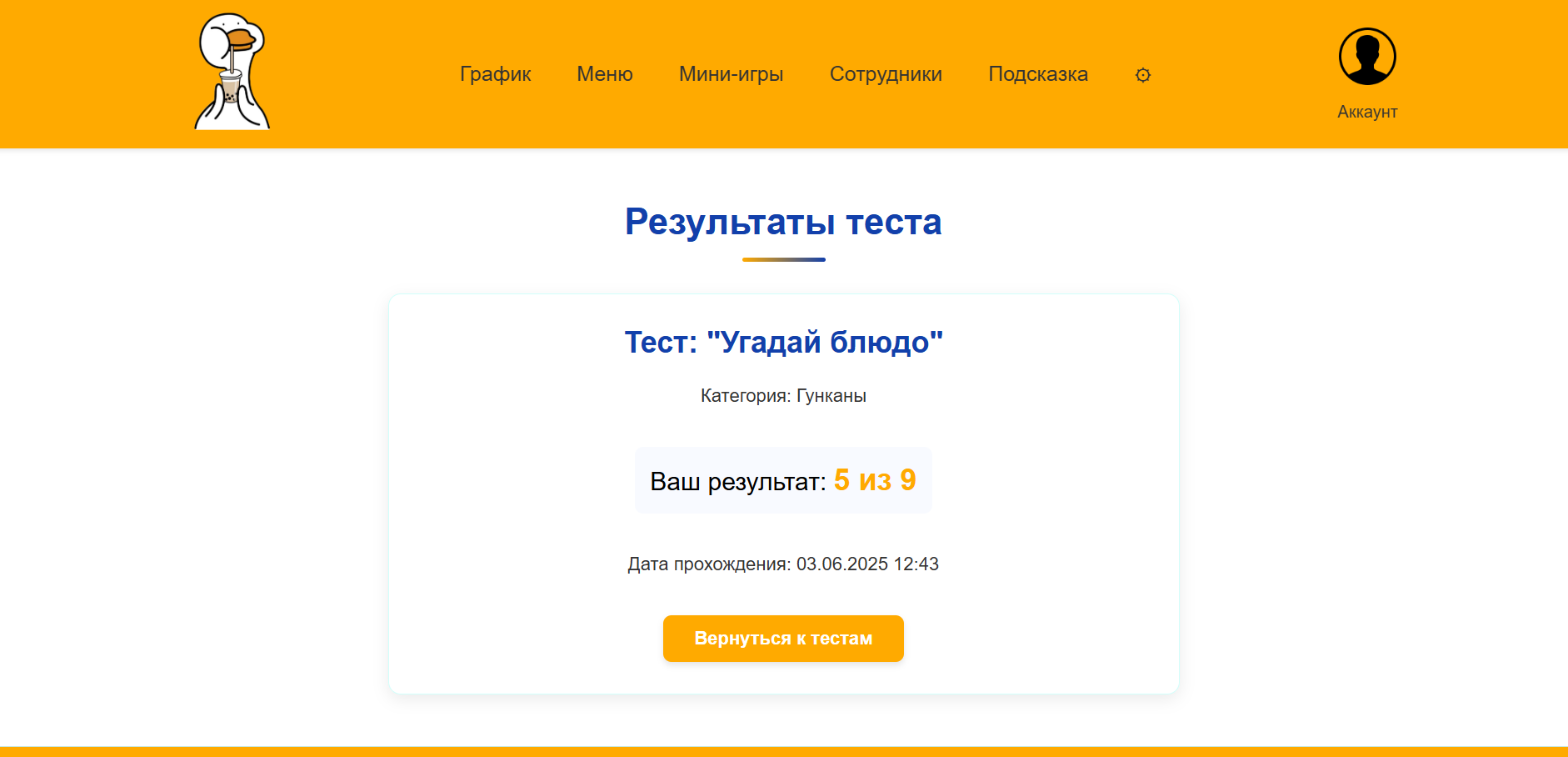


Рисунок 3.5 – Страница с результатом теста

Третьей проверкой была проверка отображения данных. Тестировщик открыл веб-приложение и просмотрел все страницы. В результате данные на всех страницах отображаются корректно, как и ожидалось. Результаты подтверждены на рисунках 3.6 - 3.12.

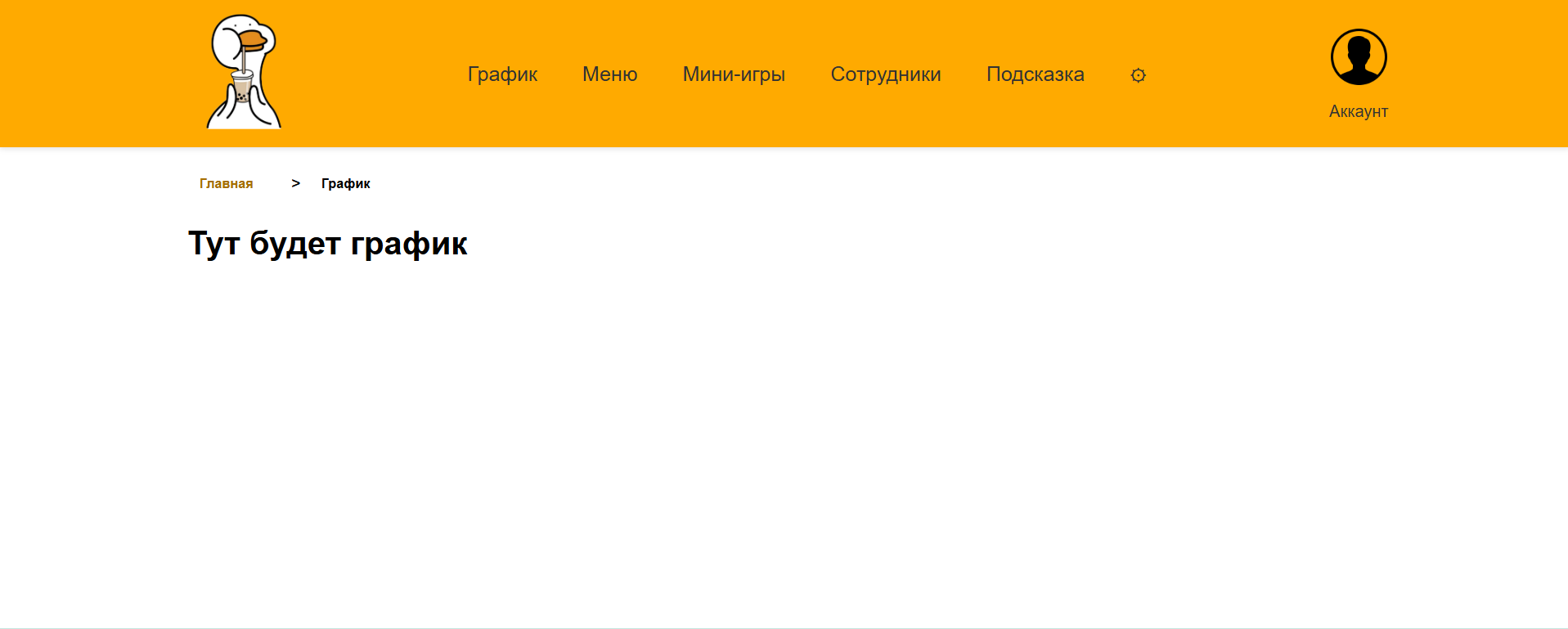


Рисунок 3.5 – Страница с графиком

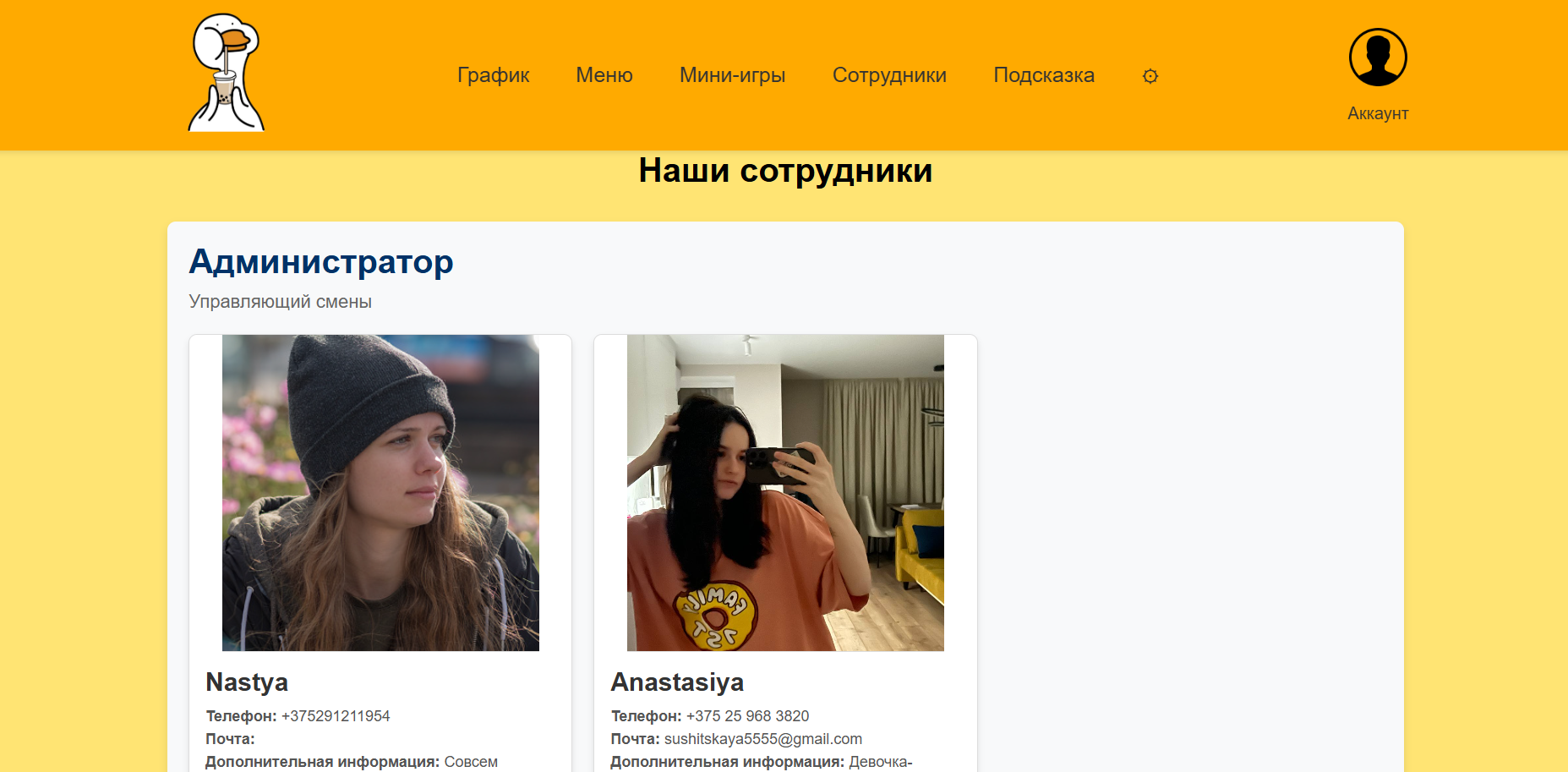


Рисунок 3.6 – Страница с информацией о сотрудниках

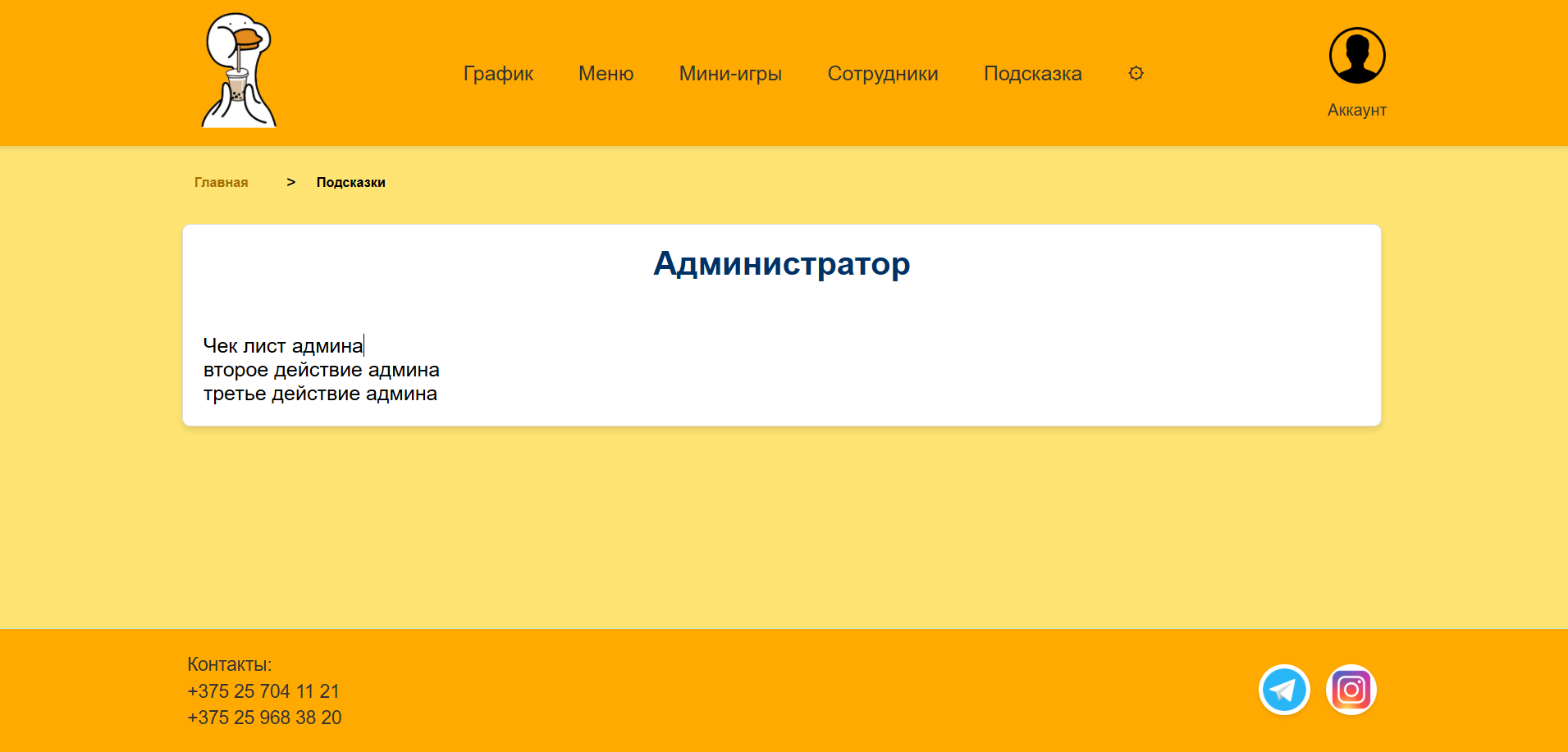


Рисунок 3.7 – Страница с чек-листами должности

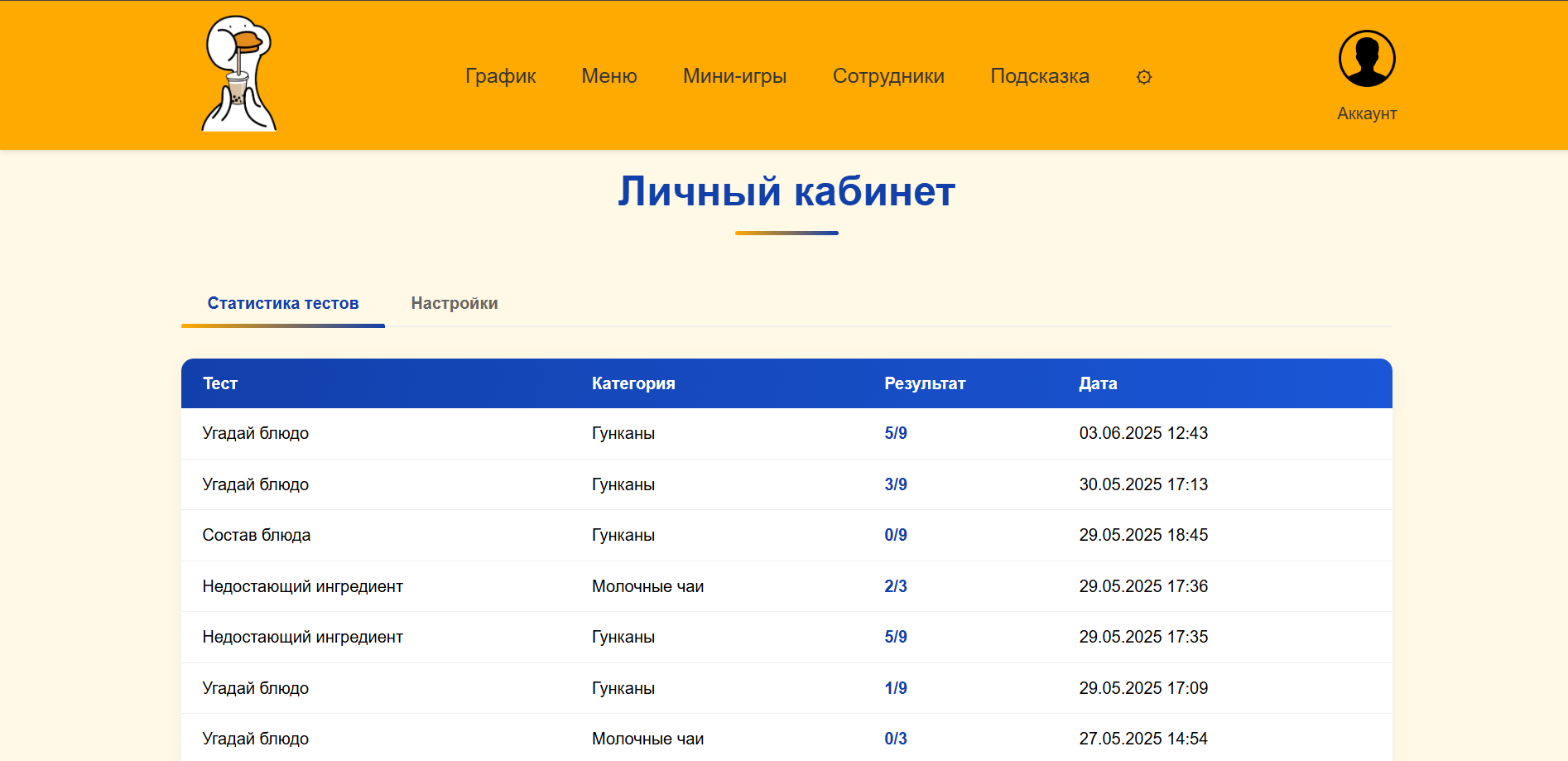


Рисунок 3.8 – Страница со статистикой тестов

Следующеё проверкой стала проверка возможностей личного аккаунта. Тетировщик выбрал в главном окне

# 4 Применение

# 4.1 Назначение веб-приложения

Главное назначение веб-приложения «HOKAIDO» – предоставить цифровой инструмент для организации, управления и прохождения стажировки в ресторане. Приложение ориентировано как на стажёров, так и на администраторов заведения. Стажёры получают доступ к необходимым обучающим материалам, расписанию занятий, заданиям и тестированию, а также могут получать уведомления, оставлять отзывы и отслеживать собственный прогресс.

Пользователь может:

* просматривать и изучать обучающие материалы;
* выполнять задания и проходить тесты;
* видеть расписание своей стажировки;
* взаимодействовать с наставником через встроенные формы обратной связи;
* отслеживать свои успехи и уровень подготовки.

Администратор взаимодействует с системой через административную панель, где доступен полный контроль над содержимым:

* добавление, редактирование и удаление обучающих модулей и тестов;
* управление новостями и расписанием;
* ведение базы стажёров и их личных кабинетов;
* отслеживание результатов прохождения программы.

Благодаря интуитивно понятному интерфейсу и продуманной логике взаимодействия, пользователи не испытывают трудностей при навигации и работе с приложением. Все элементы интерфейса соответствуют целям обучения и не отвлекают от основного содержания.

Диаграмма вариантов использования, представленная в графической части (на первом листе), наглядно иллюстрирует сценарии взаимодействия разных ролей с системой: стажёров, администраторов и наставников.

Простая и логичная структура веб-приложения обеспечивает быстрое освоение платформы даже пользователями без специальной подготовки, что особенно важно для сферы ресторанного обучения.

# 4.2 Программно-аппаратное обеспечение сервера и клиента

Для корректной работы веб-приложения, на компьютере пользователя должен быть установлен любой современный браузер.

Хостинг должен соответствовать следующим минимальным требованиям:

* предоставлять не менее 500Мб, дискового пространства;
* поддерживать программную платформу Node.js;
* предоставлять одну базу данных PostgreSQL;
* выделять не менее 2 Гб оперативной памяти;
* выделять не менее 2 Гб трафика.

Кроме этого, хостинг должен работать стабильно без перебоев и иметь канал с хорошей скоростью передачи данных. То есть, страницы должны загружаться очень быстро.

В этом не малую роль играет скорость промежуточных серверов, соединяющих их с каналом хостинга. Эти сервера обеспечивают скорость проходимости информации от хостинга к пользователю.

5 Технико-экономическое обоснование дипломного проекта

5.1 Краткая характеристика программного средства

Целью проекта является разработка веб-приложения, позволяющего стажёрам получать доступ к расписанию занятий, обучающим материалам, меню, а также взаимодействовать с наставниками и администрацией ресторана.

Использование веб-приложения позволит:

* организацию стажировки;
* заинтересовать стажёра;
* администратору отслеживать прогресс обучения стажёров;
* сотрудникам удалённо видеть график и изменения в нём;
* промотр важной информации, касающейся работы заведения

5.2 Расчет общей трудоемкости разработки программного обеспечения и трудоемкости отдельных стадий разработки

Затраты времени на разработку ПО определяются эмпирическим путем (замерами времени, фактически потраченного на выполнение отдельных стадий разработки ПО). Затраты времени включают:

* затраты труда на подготовку и описание задачи – tоп;
* затраты труда на исследование алгоритма решения задачи – tис;
* затраты труда на разработку алгоритма (блок-схем) – tал;
* затраты труда на программирование алгоритма по блок-схеме – tпр;
* затраты труда на отладку программы – tотл;
* затраты труда на подготовку документов по задаче состоят из затрат труда на подготовку и оформление документов – tд. [8]

Суммарные фактические затраты труда (∑tф) рассчитываются по формуле (5.1):

∑tф = tоп + tис + tал + tпр + tотл + tд. (5.1)

Наименование затрат времени и их объём, представлен в таблице 5.1.

Таблица 5.1 – Затраты труда

| Виды затрат труда | Обозначение | Количество чел-часов |
| --- | --- | --- |
| Затраты труда на подготовку и описание задачи | tоп | 25 |
| Затраты труда на исследование алгоритма решения задачи | tис | 20 |
| Затраты труда на разработку алгоритма (блок-схем) | tал | 35 |
| Затраты труда на программирование алгоритма по блок-схеме | tпр | 120 |
| Затраты труда на отладку программы | tотл | 50 |
| Затраты труда на подготовку и оформление документов | tд | 30 |

На основании этих данных получим:

∑tф = 25 + 20 + 35 +120 + 50 +30 = 280 (чел-часов).

Bce ПО принято подразделять на три категории сложности в соответствии с таблицей Б.1, приложения Б в зависимости от наличия (отсутствия) следующих характеристик:

* высокий уровень языкового интерфейса с пользователем;
* режим работы в реальном времени;
* управление удаленными объектами;
* машинная графика, многомашинные комплексы;
* существенное распараллеливание вычислений;
* нестандартная конфигурация технических средств;
* оптимизационные и особо сложные инженерные и научные расчеты;
* переносимость ПО.

Сравнение характеристик разрабатываемого ПО с имеющимися аналогами позволяет определить экспертным путем степень его новизны.

Программные продукты по степени новизны могут быть отнесены к одной из 4-х групп:

* группа А – разработка принципиально новых задач;
* группа Б – разработка оригинальных программ;
* группа В – разработка программ с использованием типовых решений;
* группа Г – разовая типовая задача.

Влияние фактора сложности на трудоемкость учитывается умножением фактической трудоемкости на соответствующий коэффициент сложности в зависимости от степени новизны.

Общее суммарное время на разработку (∑t) определяется по формуле (5.2):

∑t = ∑tф × Кt. (5.2)

С учетом дополнительного коэффициента сложности (Кt) рассчитаем общую трудоемкость ПС по формуле (5.2):

∑t = 280× 1,19 = 333 (чел-часов).

5.3 Расчет сметы затрат на разработку программного обеспечения

Расчет сметы затрат на разработку программного обеспечения следует делать в разрезе следующих статей:

* затраты на основную заработную плату разработчика;
* затраты на дополнительную заработную плату разработчика;
* отчисления на социальные нужды;
* отчисления в Белгосстрах;
* прочие затраты (амортизационные отчисления, расходы на электроэнергию, командировочные расходы, арендная плата за офисные помещения и оборудование, расходы на управление и реализацию, риски и т.п.).

5.3.1 Затраты на основную заработную плату разработчика

Заработная плата в коммерческих организациях Республики Беларусь предельными размерами не ограничивается и зависит от квалификации работника, его профессиональных знаний и умений. Для расчета часовой тарифной ставки (ЧТС) разработчика без опыта работы используем действующую базовую величину, а именно оплата за 1 час разработчику будет составлять 10% от действующей базовой величины в Республике Беларусь.

Часовая тарифная ставка (ЧТС) – стоимость одного часа работы. Расчет часовой тарифной ставки производится по формуле (5.3):

, (5.3)

где БВ – базовая величина, действующая в текущем году, руб. (которая равна 42 руб. с 01.01.2025 года).

Подставив данные в формулу (5.3), получим:

Расчет основной заработной платы (Зо) производится по формуле (5.4):

, (5.4)

где ∑t – суммарные затраты труда, чел-час.

Подставив данные в формулу (5.4), получим:

5.3.2 Затраты на дополнительную заработную плату разработчика

Дополнительная заработная плата (Зд) включает выплаты, предусмотренные законодательством о труде (оплата отпусков, льготных часов, времени выполнения государственных обязанностей и других выплат, не связанных с основной деятельностью исполнителей), и определяется по нормативу в процентах к основной заработной плате и составляет от 10 до 20%. Расчет дополнительной заработной платы (Зд) производится по формуле (5.5):

, (5.5)

где Зд – дополнительная заработная плата;

НД – норматив дополнительной заработной платы (%).

Подставив данные в формулу (5.5), получим:

5.3.3 Отчисления на социальные нужды

Отчисления в фонд социальной защиты населения (Зсз) определим в соответствии с действующими законодательными актами по нормативу в процентном соотношении к фонду основной и дополнительной зарплаты исполнителей, определенной по нормативу, установленному в целом по организации.

Отчисления в фонд социальной защиты населения (Зсз) определим по формуле (5.6):

, (5.6)

где Нсз – норматив отчислений в фонд социальной защиты населения 34 %.

Подставив данные в формулу (5.6), получим:

5.3.4 Отчисления в Белгосстрах

Отчисления в Белгосстрах (Збгс) определим в соответствии с действующими законодательными актами по нормативу в процентном соотношении к фонду основной и дополнительной заработной платы исполнителей. Норматив отчислений в Белгосстрах в 2025 году для данного вида деятельности установлен в размере 0,6%. Расчет отчислений в Белгосстрах (Збгс) производятся по формуле (5.7):

. (5.7)

Подставив данные в формулу (5.7), получим:

5.3.5 Прочие затраты

Расходы по статье «Прочие затраты» (Пз) на конкретное ПС включают амортизационные отчисления, расходы на электроэнергию, командировочные расходы, арендная плата за офисные помещения и оборудование, расходы на управление и реализацию, риски и т.п. Определим их по нормативу, который составляет от 10 до 20% к основной заработной плате. Расчет прочих затрат (Пз) производятся по формуле (5.8):

, (5.8)

где Нпз – норматив прочих затрат в целом по организации %.

Подставив данные в формулу (5.8), получим:

Полная сумма затрат на разработку программного обеспечения находим путем суммирования всех рассчитанных статей затрат в соответствии с таблицей 5.2.

Таблица 5.2 – Затраты на разработку программного обеспечения

|  |  |  |
| --- | --- | --- |
| Статья затрат | Условные обозначения | Сумма, руб. |
| Основная заработная плата разработчика | Зо |  |
| Дополнительная заработная плата разработчика | Зд |  |
| Отчисления на социальные нужды | Зсз |  |
| Отчисления в Белгосстрах | Збгс | 9,90 |
| Прочие затраты | Пз | 181,82 |
| Общая сумма затрат на разработку (себестоимость ПС) | Сп | 2401,19 |

5.4 Расчет экономического эффекта (прибыли) у разработчика программного обеспечения при реализации его заказчику

Прибыль по создаваемому ПС определяются исходя из результатов анализа рыночных условий, переговоров с заказчиком (потребителем) и согласования с ним отпускной цены, включающей дополнительно налог на добавленную стоимость (диапазон планируемой прибыли (рентабельности) варьирует от 15 до 40%).

Планируемую прибыль (Ппр) рассчитаем по формуле (5.9):

, (5.9)

где Ппр – прибыль от реализации ПС (руб.);

Ур – уровень рентабельности ПС (%);

Сп – себестоимость ПС (руб.).

Подставив данные в формулу (5.9), получим:

Прогнозируемую цену (Цп) без налогов рассчитываем по формуле (5.10):

Цп = Сп + Ппр. (5.10)

Подставив данные в формулу (5.10), получим:

Цп = + = 3145,56 (руб.).

Налог на добавленную стоимость (НДС) рассчитаем по формуле (5.11):

, (5.11)

где Нндс – норматив НДС – 20%.

Подставив данные в формулу (5.11), получим:

Прогнозируемая отпускная цена с учетом налогов (Цо) рассчитывается по формуле (5.12):

Цо = Цп + НДС. (5.12)

Подставив данные в формулу (5.11), получим:

Цо= + = 3774,67 (руб.).

В качестве экономического эффекта выступает чистая прибыль, которая может быть получена при реализации ПС. Чистая прибыль – это прибыль, остающаяся в распоряжении организации после уплаты налогов и обязательных отчислений.

Рассчитаем налог прибыль (Нп) от реализации ПС по формуле (5.13):

, (5.13)

где П – ставка налога на прибыль – 20%.

Подставив данные в формулу (5.13), получим:

Чистая прибыль (Пч) определяется по формуле (5.14):

Пч = Ппр – Нп. (5.14)

Подставив данные в формулу (5.14), получим:

Пч = 744,37 – 148,87 = 595,5 (руб.).

Сведем все рассчитанные выше показатели в таблицу 5.3.

Таблица 5.3 − Смета затрат и отпускная цена ПС

|  |  |  |
| --- | --- | --- |
| Статья | Обозначение | Сумма, руб. |
| Основная заработная плата разработчика | Зо |  |
| Дополнительная заработная плата разработчика | Зд |  |
| Отчисления на социальные нужды | Зсз |  |
| Отчисления в Белгосстрах | Збгс | 9,90 |
| Прочие затраты | Пз |  |
| Общая сумма затрат на разработку | Сп |  |
| Прибыль | Ппр |  |
| Прогнозируемая цена без налогов | Цп |  |
| НДС | НДС |  |
| Прогнозируемая отпускная цена | Цо | 3774,67 |
| Налог на прибыль | Нп |  |
| Чистая прибыль | Пч | 595,5 |

Вывод: разработка данного программного средства даёт экономический эффект, в виде чистой прибыли, которая будет предположительно составлять 595,5 рублей.

6 Вопросы охраны труда при работе с компьютерами

### 6.1 Безопасная организация рабочего места программиста при разработке программного обеспечения

### 6.1.1 Основные термины и определения

При разработке дипломного проекта на тему «Веб-приложение для ресторана «HOKAIDO» был рассмотрен вопрос по охране труда «Безопасная организация рабочего места программиста при разработке программного обеспечения»:

Охрана труда – это система законодательных, нормативных, организационных, технических, санитарно-гигиенических, лечебно-профилактических и культурно-просветительных мероприятий, направленных на обеспечение безопасных и комфортных условий труда, сохранение здоровья работников и предупреждение производственных травм и профессиональных заболеваний.

Целью охраны труда является предотвращение производственных травм и профессиональных заболеваний, обеспечение безопасных и комфортных условий труда, улучшение качества жизни работников и повышение эффективности производственной деятельности. Кроме того, целью охраны труда является сокращение экономических потерь, связанных с производственными травмами и заболеваниями, а также соблюдение законодательных требований в области охраны труда и социальной защиты работников.

Современная индустрия информационных технологий активно развивается, и профессия программиста становится одной из самых востребованных. Однако длительная работа за компьютером сопряжена с рядом профессиональных рисков, которые могут негативно сказаться на здоровье специалиста. Поэтому важным аспектом профессиональной деятельности является организация безопасных условий труда, включающая правильное оборудование рабочего места, соблюдение режимов работы и отдыха, а также создание психологически комфортной среды.

6.1.2 Основные принципы организации безопасного рабочего места

6.1.2.1. Эргономика рабочего пространства

Эргономика — это наука о проектировании рабочих мест с учетом физиологических особенностей человека. Правильная эргономика способствует снижению утомляемости, профилактике профессиональных заболеваний и повышению эффективности труда.

Мебель: рабочий стол должен иметь регулируемую высоту (обычно 70-80 см), чтобы обеспечить комфортное положение рук и спины. Стул — с регулируемой высотой, с поддержкой поясницы и возможностью регулировки угла наклона спинки. Важно обеспечить устойчивость мебели и отсутствие люфтов.

Рабочая зона: монитор располагается на уровне глаз или чуть ниже, примерно в 50-70 см от глаз. Расстояние до экрана должно позволять видеть его полностью без напряжения глаз. Клавиатура и мышь — так, чтобы руки находились в нейтральном положении, локти — под углом около 90 градусов.

Организация пространства: все необходимые предметы должны быть под рукой, чтобы не приходилось часто наклоняться или вытягиваться. Это снижает риск травм и повышает комфорт.

6.1.2.2. Освещение

Правильное освещение — залог здоровья глаз и общего самочувствия:

Предпочтительно использовать естественный свет, избегая прямых солнечных лучей на экран.

Искусственное освещение должно быть мягким, равномерным, без резких теней и бликов, КЕО не должно быть ниже 1,5%.

Освещённость на поверхности стола в зоне размещения документов должна быть 400 лк (при люминесцентном освещении).

Экран компьютера должен иметь антибликовое покрытие или защитные фильтры, на экране ВДТ должен быть 200лк.

6.1.2.3. Микроклиматические условия

Оптимальный температурный режим (19–21°C), влажность воздуха (62–55%), уровень шума не выше 65 дБА и хорошая вентиляция (скорость движения воздуха - 0,1) создают комфортные условия для работы:

Регулярное проветривание помещения.

Использование очистителей воздуха при необходимости.

Избегание сквозняков и чрезмерной сухости воздуха.

6.1.2.4. Организация режима труда и отдыха

Длительная работа без перерывов негативно влияет на здоровье:

Рекомендуется работать не более 1–2 часов подряд без перерыва.

Каждые 45–60 минут необходимо делать короткую паузу (5–10 минут) для разминки глаз (например, упражнений по методу Бейтса), растяжки или прогулки.

В течение дня важно чередовать работу за компьютером с физическими упражнениями или отдыхом.

6.1.2.5. Психологическая безопасность

Создание психологически комфортных условий способствует снижению стрессов:

Обеспечение спокойной рабочей обстановки.

Ограничение отвлекающих факторов (шум, посторонние разговоры).

Возможность индивидуальной настройки рабочего пространства под личные предпочтения.

6.1.2.6. Защита зрения

Длительная работа за компьютером может привести к развитию синдрома компьютерного зрения:

Регулярные перерывы для глаз.

Использование специальных очков с антибликовым покрытием.

Настройка яркости и контрастности экрана в соответствии с освещением помещения.

6.1.2.7. Безопасность информационной среды

Обеспечение защиты данных и предотвращение киберугроз также входит в аспекты безопасной работы:

Использование антивирусных программ.

Регулярное обновление программного обеспечения.

Создание резервных копий данных.

6.1.3 Потенциальные профессиональные риски при работе программиста

Несмотря на кажущуюся простоту работы за компьютером, у специалистов могут развиваться следующие профессиональные заболевания:

Синдром компьютерного зрения — усталость глаз, сухость слизистых оболочек, головные боли.

Боли в спине и шее — вследствие неправильной осанки или неудобной мебели.

Туннельный синдром запястья — при неправильном положении кистей во время работы с клавиатурой или мышью.

Психоэмоциональное переутомление — стрессовые ситуации или монотонная работа.

Меры профилактики профессиональных заболеваний

Для снижения риска развития вышеуказанных заболеваний необходимо соблюдать следующие рекомендации:

Регулярно проводить разминку глаз (например, по методу Бейтса).

Следить за правильной осанкой во время работы.

Использовать эргономичную мебель и аксессуары (подставки для ног, подушки под поясницу).

Обеспечивать правильное освещение помещения.

Делать перерывы для физических упражнений или прогулок на свежем воздухе.

Ограничивать время работы за компьютером в вечернее время для профилактики нарушения сна.

Организация безопасных условий труда при работе за компьютером является важным аспектом охраны труда, особенно для программистов, чья деятельность связана с длительным сидением и использованием электронных устройств. Эффективная профилактика профессиональных заболеваний достигается за счет соблюдения принципов эргономики, правильного освещения, оптимальных микроклиматических условий, режима труда и отдыха, а также психологической поддержки. Важными мерами являются регулярные перерывы, упражнения для глаз и тела, использование эргономичной мебели и средств защиты зрения. Кроме того, необходимо обеспечивать информационную безопасность и защиту данных. Соблюдение этих рекомендаций способствует сохранению здоровья специалистов, повышению их работоспособности и снижению экономических потерь, связанных с профессиональными заболеваниями и травмами.

# 7 Энерго- и ресурсосбережение

В Республике Беларусь регулирование в области ресурсо- и энергосбережения осуществляется в рамках Государственной программы "Энергосбережение" на 2021-2025 годы, утвержденной постановлением Совета Министров от 24.02.2021 № 103. Основные цели программы - снижение импортозависимости, повышение эффективности использования ресурсов, использование возобновляемых источников энергии и внедрение энергоэффективных технологий.

Государственная программа Беларуси по энергосбережению направлена на снижение зависимости экономики от импорта энергоресурсов и повышение энергоэффективности. За 2016-2020 годы удалось добиться значительных результатов: энергоемкость ВВП снизилась более чем в 2 раза по сравнению с 2000 годом, достигнув показателей развитых стран. Общий объем экономии топливно-энергетических ресурсов составил 4,1 млн тонн условного топлива, в том числе за счет внедрения возобновляемых источников энергии (ВИЭ), доля которых в энергобалансе страны увеличилась до 7,1%.

Ключевыми достижениями стали ввод в эксплуатацию 106 энергоисточников на местных ресурсах, 17 солнечных электростанций, 4 ГЭС, 36 ветроустановок и 14 биогазовых комплексов. Программа также включала масштабную просветительскую работу через конкурсы, публикации и образовательные мероприятия. Однако для дальнейшего снижения энергозависимости необходимо продолжать внедрение энергоэффективных технологий, развитие ВИЭ и систем энергоменеджмента во всех отраслях экономики.

Государственная программа Беларуси по энергосбережению на 2021-2025 годы ставит две ключевые цели: сдерживание роста потребления топливно-энергетических ресурсов (ТЭР) при экономическом развитии страны и увеличение доли местных энергоресурсов, включая возобновляемые источники (ВИЭ). Для их достижения предусмотрены две подпрограммы: по повышению энергоэффективности (экономия 2,5-3 млн т у.т.) и развитию местных ТЭР (доведение их доли до 16,1%, а ВИЭ - до 7-8% к 2026 году).

Основной целевой показатель - снижение энергоемкости ВВП на 7% к 2026 году при росте экономики на 21,5%. Программа согласована с Целями устойчивого развития ООН и реализуется под руководством Госстандарта при участии 20 министерств и ведомств, включая отраслевые концерны и местные органы власти. Мероприятия носят межотраслевой характер и интегрированы в другие госпрограммы.

Источниками финансирования мероприятий Государственной программы являются средства республиканского и(или) местных бюджетов (в том числе предусмотренные на финансирование Государственной программы), собственные средства организаций, кредитные ресурсы банков Республики Беларусь, открытого акционерного общества «Банк развития Республики Беларусь», иные источники, не запрещенные законодательством (в том числе средства внебюджетных инвестиционных фондов, международных финансовых организаций, гранты, иностранные инвестиции).

Для финансирования общего комплекса мероприятий за счет всех источников потребовалось 4497874951,7 рубля, в том числе в 2021 году – 808422000 рублей, в 2022 году – 962683000 рублей, в 2023 году – 739332000 рублей, в 2024 году – 961043596,7 рубля и потребуется в 2025 году – 1026394355 рублей.

Закупка отечественного оборудования осуществляется при реализации в рамках Государственной программы мероприятий по строительству энергоисточников на местных ТЭР по результатам проведения процедур закупок в соответствии с законодательством Республики Беларусь или международными правилами проведения закупок, применяемыми при реализации международных проектов.

Капитальные вложения по перечню объектов по внедрению тепловых насосов в соответствии с показателями ввода в эксплуатацию энергоисточников на местных ТЭР будут обеспечиваться в рамках региональных комплексов мероприятий (региональных программ) и региональных инвестиционных программ.

Реализация Государственной программы по энергосбережению может столкнуться с рядом рисков, требующих особого внимания и управления. Среди основных рисков выделяются макроэкономические факторы, включая возможное замедление темпов экономического роста, снижение производственных показателей в различных секторах экономики, уменьшение инвестиционной активности предприятий, а также рост стоимости топливно-энергетических ресурсов и тарифов на энергию. Значительные финансовые риски связаны с возможной нехваткой или несвоевременным поступлением финансирования, высокой закредитованностью организаций-исполнителей и недостатком у них оборотных средств. Отдельную категорию составляют правовые риски, обусловленные изменениями в законодательной базе.

Для эффективного управления этими рисками и минимизации их последствий разработан комплекс мер. Он включает регулярную корректировку целевых показателей программы с учетом изменяющихся макроэкономических условий, своевременный учет планируемых законодательных изменений и тщательный отбор исполнителей мероприятий. Особое внимание уделяется вопросам финансирования - обеспечению своевременного и эффективного использования средств, привлечению дополнительных внебюджетных источников, ежегодному уточнению объемов финансирования в соответствии с экономической ситуацией. Важным элементом является определение приоритетов для первоочередного финансирования и возможность перераспределения средств между наиболее важными направлениями программы.

Система управления рисками также предусматривает постоянный мониторинг и оценку результатов реализации подпрограмм, организацию информационного обеспечения, изучение зарубежного опыта и обмен лучшими практиками. Особое значение придается кадровому обеспечению через подготовку, переподготовку и повышение квалификации специалистов, участвующих в выполнении программы. В случаях, когда отдельные мероприятия не будут выполнены в установленные сроки, предусмотрена возможность их завершения в рамках следующей пятилетней программы по энергосбережению, что обеспечивает преемственность и непрерывность работы в этой сфере. [16]

# ЗАКЛЮЧЕНИЕ

В проекте разработано веб-приложение «HOKAIDO», предназначенного для автоматизации процесса стажировки новых сотрудников в ресторане

В приложении реализованы следующие функции:

* реализация системы ролей: стажёр, наставник, администратор;
* разработка личного кабинета стажёра с расписанием, материалами, заданиями и обратной связью;
* организация панели администратора с возможностью добавления, редактирования и удаления курсов, расписаний и пользователей;
* обеспечение фильтрации данных (по сменам, наставникам, дате);
* создание базы готовых обучающих программ и модулей.

В целом, поставленные задачи были выполнены в полном объёме, и приложение может использоваться на практике для стажировки сотрудников в ресторане, повышая эффективность подготовки кадров и снижая нагрузку на наставников и менеджеров.

В процессе работы над проектом были закреплены навыки проектирования баз данных, создания интерфейсов, обработки пользовательского ввода, а также реализована безопасная работа с персональными данными. Были применены знания в области веб-разработки, Python и Django.

В проекте также рассмотрен вопрос по охране труда: «Безопасная организация рабочего места программиста при разработке программного обеспечения».

Выполнена графическая часть проекта, включая диаграмму вариантов использования, диаграмму классов и схему базы данных.

Разработка данного программного средства даёт экономический эффект, в виде чистой прибыли, которая будет предположительно составлять 595,5 рублей.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Инфорсер [Электронный ресурс]. – Режим доступа: https://roscomputer.ru/products/rol-kompyuterov-v-sovremennom-mire. – Дата доступа: 02.04.2024.
2. Allbest [Электронный ресурс]. – Режим доступа: https://otherreferats.allbest.ru/programming/01127677\_0.html. – Дата доступа: 02.04.2024.
3. Выбор ПК [Электронный ресурс]. – Режим доступа: https://works.doklad.ru/view/rJqwx0kj-pM.html. – Дата доступа: 07.05.2023.
4. Ram.by [Электронный ресурс]. – Режим доступа: https://ram.by/. – Дата доступа: 02.04.2024.
5. Cooler.by [Электронный ресурс]. – Режим доступа: https://cooler.by/. – Дата доступа: 02.04.2024.
6. WebStorm [Электронный ресурс]. – Режим доступа: https://www.jetbrains.com/ru-ru/webstorm/. – Дата доступа: 02.04.2024.
7. TypeScript [Электронный ресурс]. – Режим доступа: https://metanit.com/web/typescript/1.1.php. – Дата доступа: 02.04.2024.
8. PostgreSQL [Электронный ресурс]. – Режим доступа: https://www.postgresql.org/about/. – Дата доступа: 02.04.2024.
9. Peaksoft [Электронный ресурс]. – Режим доступа: https://www.apeaksoft.com/ru/mind-map/review-drawio.html. – Дата доступа: 02.04.2024.
10. Mui.com [Электронный ресурс]. – Режим доступа: https://mui.com/. – Дата доступа: 02.04.2024.
11. MobX [Электронный ресурс]. – Режим доступа: https://mobx.js.org/README.html. – Дата доступа: 02.04.2024.
12. NestJS [Электронный ресурс]. – Режим доступа: https://docs.nestjs.com/. – Дата доступа: 02.04.2024.
13. Next.js [Электронный ресурс]. – Режим доступа: https://nextjs.org/docs. – Дата доступа: 02.04.2024.
14. Сокол, Т.С. Охрана труда: Учебное пособие / под общ. ред. Н. В. Овчинниковой. – Мн.: Дизайн ПРО, 2006. – 166 с.
15. Макаров, Л.К. Пожарная безопасность: Энциклопедия / редактор издательства Ю.В. Полежаева. – М.: Голден-Би, 2007. – 76 с.
16. StudFiles [Электронный ресурс]. – Режим доступа: https://studfile.net/preview/5633601/. – Дата доступа: 02.04.2024.

# ПРИЛОЖЕНИЕ А

(справочное)

Фрагменты текста программы

from django import forms

from django.contrib.auth.forms import UserCreationForm

from .models import Account, Specialist, Service

import datetime

class SignupForm(UserCreationForm):

class Meta:

model = Account

fields = ['last\_name', 'first\_name', 'middle\_name', 'date\_of\_birth', 'phone', 'email', 'password1', 'password2']

def clean\_email(self):

email = self.cleaned\_data.get('email')

if Account.objects.filter(email=email).exists():

raise forms.ValidationError("Этот адрес электронной почты уже используется.")

return email

def \_\_init\_\_(self, \*args, \*\*kwargs):

super().\_\_init\_\_(\*args, \*\*kwargs)

for field\_name, field in self.fields.items():

if self.initial.get(field\_name) is None:

self.initial[field\_name] = '' # Заменяем None на пустую строку

class BookingForm(forms.Form):

specialist = forms.ModelChoiceField(queryset=Specialist.objects.all(), label="Специалист")

service = forms.ModelChoiceField(queryset=Service.objects.all(), label="Услуга")

date = forms.DateField(

label="Дата",

widget=forms.DateInput(attrs={"type": "date", "min": datetime.date.today().isoformat()}),

required=False

)

def \_\_init\_\_(self, \*args, fixed\_specialist=None, fixed\_service=None, \*\*kwargs):

super().\_\_init\_\_(\*args, \*\*kwargs)

if fixed\_specialist:

self.fields['specialist'].queryset = Specialist.objects.filter(id=fixed\_specialist.id)

self.fields['specialist'].initial = fixed\_specialist

self.fields['specialist'].disabled = True

self.fields['service'].queryset = Service.objects.filter(

specialists\_\_specialist=fixed\_specialist

)

if fixed\_service:

self.fields['service'].queryset = Service.objects.filter(id=fixed\_service.id)

self.fields['service'].initial = fixed\_service

self.fields['service'].disabled = True

self.fields['specialist'].queryset = Specialist.objects.filter(

services\_\_service=fixed\_service

)

from django import forms

from django.contrib.auth import get\_user\_model

from django.core.exceptions import ValidationError

User = get\_user\_model()

class UserProfileForm(forms.ModelForm):

class Meta:

model = User

fields = ['first\_name', 'last\_name', 'middle\_name', 'email', 'phone']

widgets = {

'first\_name': forms.TextInput(attrs={'class': 'form-input'}),

'last\_name': forms.TextInput(attrs={'class': 'form-input'}),

'middle\_name': forms.TextInput(attrs={'class': 'form-input'}),

'email': forms.EmailInput(attrs={'class': 'form-input'}),

'phone': forms.TextInput(attrs={'class': 'form-input'}),

}

def clean\_phone(self):

phone = self.cleaned\_data['phone']

# Можно добавить валидацию номера телефона

if len(phone) < 5: # Пример простой проверки

raise ValidationError("Номер телефона слишком короткий")

return phoneimport styles from './styles.module.scss';

# ПРИЛОЖЕНИЕ Б

(обязательное)

Дополнительная информация по экономическому разделу

Таблица Б.1 – Характеристики категорий сложности ПО

|  |  |  |  |
| --- | --- | --- | --- |
| Категории сложности | Характеристики ПО | Степень новизны | Дополнительные коэффициенты сложности (Кt) ПО |
| 1 | ПО, обладающие одной из следующих характеристик: наличие сложного интеллектуального языкового интерфейса с пользователем; обеспечение телекоммуникационной обработки данных и управление удаленными объектами; обеспечение существенного распараллеливания вычислений;  криптография и другие методы защиты информации | А | 1,38 |
| Б | 1,26 |
| В | 1,15 |
| Г | 0,69 |
| 2 | ПО, обладающие одной или несколькими из следующих характеристик: моделирование объектов и процессов; обеспечение настройки ПО на изменения | А | 1,30 |
| Б | 1,19 |
| В | 1,08 |
| Г | 0,65 |
| 3 | ПО, не обладающие перечисленными выше характеристиками. | А | 1,20 |
| Б | 1,10 |
| В | 1,00 |
| Г | 0,60 |

Варицкая А.В.

Варицкая А.В.

ААА.В.

Сушицкая А.П.

Утв.

Н. контр.

Т. контр.

Провер.

Разраб.

*Изм*

*Лист*

*№ докум*

*Подп.*

.

*Дата*

Т

.

.

Н

.

.

Утв

.

*Лит*

.

*Лист*

*Листов*

-

«

»

*Ведомость дипломного проекта*

гр

9

*47*

9

4

*МГКЦТ ДП63ТП.005109.081 ПЗ*

1

-

08

1

*Обозначение*

*Наименование*

*Дополнитель-*

*ные сведения*

1

ПЗ

*Текстовые документы*

*Пояснительная записка*

с

.

*Отзыв руководителя*

1

с

.

*Рецензия*

1

с

.

*Графические документы*

ГУИР

.

63111

2

.

001

ПЛ

*Диаграмма вариантов использования*

*Формат А3*

ГУИР

.

6311

12

.

002

ПЛ

1

ГУИР

.

63111

2

.

003

ПЛ

1

ГУИР

.

63111

2

.

004

ПЛ

ГУИР

.

63111

2

.

005

ПЛ

ГУИР

.

63111

2

.

006

ПЛ

1

Инв

.

подл

.

Подп

.

и дата

Взам

.

инв

.

Инв

.

дубл

.

Подп

.

и дата

Перв

.

примен

.

Справ

.

*Электронный носитель информации с*

*программным обеспечением и материалами*

HLD

625

UI

CD

-

RW

Шавейко А.А.

Левонюк Е.А.

*Диаграмма классов*

*USB-flash*

*УО «МГКЦТ»*

*47*

*МГКЦТ* *ДП63ТП.022109*

*1 л.*

*1 с.*

*47 с.*

*Разработка веб-приложения для ресторана «HOKAIDO»*

*Формат А3*

*Формат А3*

*МГКЦТ ДП63ТП.022109.081ПЗ*

*Схема базы данных*

*МГКЦТ ДП63ТП.022109.003 Д*

*МГКЦТ ДП63ТП.022109.002 Д*

*МГКЦТ ДП63ТП.022109.001 Д*