**ПРИЛОЖЕНИЕ А ТЕХНИЧЕСКОЕ ЗАДАНИЕ**

МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

им. Н.Э. Баумана

Кафедра «Системы обработки информации и управления»

|  |  |  |
| --- | --- | --- |
| Утверждаю  Заведующий кафедрой ИУ-5 |  | Согласовано  Научный руководитель |
| \_\_\_\_\_\_\_\_\_\_\_В.И.Терехов  "\_\_"\_\_\_\_\_\_\_\_\_\_2024 г. |  | \_\_\_\_\_\_\_\_\_\_А.И. Канев  "\_\_"\_\_\_\_\_\_\_\_\_\_2024 г. |

**Консалтинг по ИТ безопасности**

Техническое задание

(вид документа)

писчая бумага

(вид носителя)

8

(количество листов)

|  |  |
| --- | --- |
| ИСПОЛНИТЕЛЬ: |  |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Гонов Марат |
| "\_\_"\_\_\_\_\_\_\_\_\_\_\_2024 г. |  |

Москва - 2024

1. Введение

Система «Consulting» представляет собой веб-сервис, включающий в себя веб-приложение и нативное приложение для организации процесса оказания консультационных услуг в сфере информационной безопасности, управления заявками и взаимодействия между консультантами и клиентами.

1. Назначение разработки

Данная система предназначена для автоматизации полного цикла оказания консультационных услуг в сфере информационной безопасности. С её помощью пользователи (клиенты) могут формировать заявки на консультации, выбирать необходимые услуги из каталога и получать экспертные услуги по защите корпоративной сети, а консультанты — оперативно обрабатывать эти заявки, планировать график консультаций и формировать отчёты о проделанной работе. Таким образом, система способствует оптимизации внутренних процессов, контролю исполнения задач по кибербезопасности и более эффективному взаимодействию между клиентами и специалистами.

1. Стадии и этапы разработки
   1. Создание MVP и базового дизайна на основе infosecurity
   2. Создание базы данных для хранения информации о получателях и отправках файла в PostgreSQL
   3. Создание веб-сервиса на DRF
   4. Реализация авторизацию и хранение отозванных сессий в Redis
   5. Разработка SPA на React для гостя
   6. Внедрение адаптивности, менеджера состояний Redux Toolkit, PWА, разработка Tauri приложения
   7. Завершение разработки интерфейса отправителя на React
   8. Добавить в приложение интерфейс получателя на React
   9. Разработка десктопного приложения на Tauri
   10. Развертывание приложения при помощи GitHub Pages
   11. Подготовка всей документации (РПЗ, ТЗ и набор диаграмм)
   12. Оформление git-репозитория на сервисе GitHub
   13. Анализ сервисов-аналогов
2. Требования к функциональным характеристикам
   1. Методы HTTP
      1. GET Получение всех услуг
      2. POST Добавление услуги
      3. GET Получение услуги
      4. PUT Изменение услуги
      5. DELETE Удаление услуги
      6. POST Добавление услуги в заявку
      7. POST Изменение/добавление картинки заявки
      8. GET Получение списка всех заявок
      9. GET Получение заявки
      10. PUT Обновление полей заявки
      11. DELETE Удаление услуги из заявки
      12. PUT Формирование заявки
      13. PUT Завершение заявки
      14. DELETE Удаление заявки
      15. PUT Изменение комментария для услуги в заявке
      16. POST Регистрация пользователя
      17. PUT Изменение данных пользователя (Личный кабинет)
      18. POST Аутентификация
      19. POST Деавторизация
   2. Меню
      1. Consulting - перенаправляет на страницу 4.6
      2. Услуги – перенаправляет на страницу 4.7
      3. Заявки – перенаправляет на страницу 4.10
      4. Выпадающее меню
         1. Изменить профиль – перенаправляет на страницу 4.5
         2. Выход – (вызывается метод 4.1.19)
   3. Регистрация
      1. Доступно только гостям
      2. Отображает форму регистрации
         1. Поле логина пользователя
         2. Поле пароля
      3. Действия
         1. Регистрация пользователя – (вызывается метод 4.1.16)
         2. Войти – перенаправляет на страницу 4.4
   4. Аутентификация
      1. Доступно только гостям
      2. Отображает форму аутентификации
         1. Поле логина
         2. Поле пароля
      3. Действия
         1. Войти – (вызывается метод 4.1.18)
         2. Регистрация – перенаправляет на страницу 4.3
   5. Личный кабинет
      1. Доступно аутентифицированному пользователю
      2. Действия
         1. Изменить данные пользователя – (вызывается метод 4.1.17)
   6. Главная
      1. Доступна всем
      2. Отображается статическая информация о сервисе
         1. Назначение сервиса
   7. Список услуг
      1. Доступна всем
      2. Отображаются элементы карточек с услугами, вызывается метод 4.1.1
         1. Картинка
         2. Цена
      3. Действия
         1. Поиск – перенаправляет на страницу 4.7, (используется метод 4.1.1), с фильтрующем параметром
         2. Подробнее – перенаправляет на страницу 4.8 (используется метод 4.1.3)
         3. Добавить – добавляет получателя в заявку-черновик, (вызывается метод 4.1.2), только аутентифицированные пользователи.
         4. Корзина – перенаправляет на страницу 4.9, только аутентифицированные пользователи.
   8. Одна услуга
      1. Доступна всем
      2. Отображается подробная информация о выбранной услуге, (вызывается метод 4.1.3)
   9. Одна заявка
      1. Доступно только аутентифицированным посетителям
      2. Отображает текущий заказ-черновик пользователя, (метод 4.1.9)
         1. Список выбранных услуг
         2. Поле комментария к услуге
      3. Действия, доступны только в случае, если статус «черновик»
         1. Удалить услугу – удалят услугу из отправки, (вызывается метод 4.1.11)
         2. Сохранить – сохраняет комментарий для услуги, (вызывается метод 4.1.15)
         3. Удалить заявку – удаляет заявку-черновик, (вызывается метод 4.1.14)
         4. Заполнить – вносит данные заявки (вызывается метод 4.1.10)
   10. Заявки
       1. Доступно аутентифицированному посетителю
       2. Отображается список заявок
          1. Только заявки, созданные данным пользователем, если открыл пользователь
          2. Все оформленные заявки, если открыл администратор
       3. Действия
          1. Фильтрация – фильтрует заявки по дате создания и статусу, вызывается (метод 4.1.8)
          2. Сформировать – формирует заявку, выполняется метод 4.1.13, доступно только пользователю
          3. Отклонить – отклоняет заявку, вызывается метод 4.1.13, доступно только администратору
          4. Посмотреть подробную информацию об услуге – перенаправляет на страницу 4.9 (вызывается метод 4.1.9)
   11. Список услуг таблицей
       1. Доступно только администратору
       2. Отображаются все существующие в системе услуги (вызывается метод 4.1.1)
       3. Действия
          1. Удалить – удаляет услугу (вызывается метод 4.1.5)
          2. Редактирование/создание – переход на страницу 4.12
   12. Редактирование/создание услуг
       1. Доступно только администратору
       2. Отображается информация об услуге (вызывается метод 4.1.3)
          1. Название
          2. Описание
          3. Статус
          4. Цена
          5. Время выполнения
          6. Изображение
       3. Действия
          1. Сохранить – сохранение изменений в существующей услуге (вызывается метод 4.1.4)
          2. Добавить – добавление новой услуги (вызывается метод 4.1.2)
   13. 404
       1. Доступно всем
       2. Отображается в случае отсутствия ресурса
       3. Действия.
          1. Вернуться на главную – переход на страницу 4.6.
   14. 403
       1. Доступно всем
       2. Отображается в случае отсутствия прав на использование ресурса
       3. Действия.
          1. Вернуться на главную – переход на страницу 4.6.
3. Требования к составу и параметрам технических средств
   1. Сервер
      1. Процессор AMD Ryzen 5
      2. Оперативная память 16 Гб
      3. Свободное пространство на диске 10 Гб
   2. Клиент
      1. Процессор AMD Ryzen 5
      2. Оперативная память 16 Гб
      3. Свободное пространство на диске 10 Гб
4. Требования к информационной и программной совместимости
   1. Сервер
      1. ОС Ubuntu (24.10 и выше)
      2. Minio (RELEASE.2023-12-14T18-51-57Z)
      3. Redis (7.2)
      4. Node JS (1.23)
      5. PostgreSQL (17)
      6. Python 3.12
      7. Доступ к интернету
   2. Клиент
      1. Браузер (Safari 16.5.2, Firefox 121.0, Chrome 119.0.6045, Yandex 24.6.3.729, Opera 105.0.4970.16)

**ПРИЛОЖЕНИЕ Б СПИСОК HTTP МЕТОДОВ**

Таблица 2 – HTTP методы разрабатываемого веб-сервиса

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№** | **Тип** | **URL** | **Описание** | **Входные данные** | **Выходные данные** |
| 4.1.1 | GET | / services/ | Возвращает список активных услуг Доступно всем | name=string | {  "draft\_request\_id": int, "services\_in\_draft\_request": int,  "services":  [  {  "id": int,  "name": string,  "description": string,  "price": float,  "status": string,  "image\_url": string  }  ]  } |
| 4.1.2 | POST | /services/create/ | Создаёт новую услугу | {  "name": string,  "description": string,  "price": float,  "status": string,  "image\_url": string  } | {  "id": int,  "name": string,  "description": string,  "price": float,  "status": string,  "image\_url": string  } |
| 4.1.3 | GET | /services/{pk}/ | Возвращает информацию об услуге по ID | pk: int | {  "id": int,  "name": string,  "description": string,  "price": float,  "status": string,  "image\_url": string  } |
| 4.1.4 | PUT | /services/{pk}/update/ | Обновляет данные услуги, доступно менеджеру | pk: int  {  "name": string,  "description": string,  "price": float,  "status": string,  "image\_url": string  } | {  "id": int,  "name": string,  "description": string,  "price": float,  "status": string,  "image\_url": string  } |
| 4.1.5 | DELETE | /services/{pk}/  delete/ | Удаляет услугу | pk: int | {  "message": "Service deleted"  } |
| 4.1.6 | POST | /services/{pk}/  add-to-draft/ | Добавляет услугу в черновую заявку | pk: int | {  "request\_id": int,  "service\_id": int,  "quantity": int  } |
| 4.1.7 | POST | /services/  {service\_id}/  update-image/ | Обновляет изображение для услуги | pk: int  {  "service\_id": int,  "image": Image } | {  "detail": string,  "image\_url": string  } |
| 4.1.8 | GET | /requests/ | Возвращает список заявок | start\_date (опционально): string, end\_date (опционально): string, status (опционально): string | [  {  "id": int,  "status": string,  "creation\_date": datetime,  "total\_cost": float,  "client": string,  "services":  [  {  "id": int,  "name": string,  "price": float,  "quantity": int  }  ]  }  ] |
| 4.1.9 | GET | /requests/{pk}/ | Возвращает информацию о заявке по ID | pk: int | {  "id": int,  "status": string,  "creation\_date": datetime,  "completion\_date": datetime,  "total\_cost": float,  "client": string,  "services":  [  {  "id": int,  "name": string,  "price": float,  "quantity": int  }  ]  } |
| 4.1.10 | PUT | /requests/{pk}/  update/ | Обновляет заявку | pk: int  {  "status": string,  "total\_cost": float  } | {  "id": int,  "status": string,  "creation\_date": datetime,  "completion\_date": datetime,  "total\_cost": float,  "client": string,  "services":  [  {  "id": int,  "name": string,  "price": float,  "quantity": int  }  ]  } |
| 4.1.11 | DELETE | /requests/{pk}/  delete/ | Удаляет заявку | pk: int | {  "id": int,  "status": string,  "message": "Request successfully deleted."  } |
| 4.1.12 | PUT | /requests/{pk}/  form/ | Переводит черновую заявку в статус "Submitted" | pk: int | {  "message": "Request successfully submitted.", "status": string,  "submission\_date": datetime  } |
| 4.1.13 | PUT | /requests/{pk}/  complete-or-reject/ | Меняет статус заявки ("Completed" или "Rejected") | pk: int  {  status: string } | {  "message": "Request successfully completed/rejected.",  "status": string,  "completion\_date": datetime,  "total\_cost": float  } |
| 4.1.14 | DELETE | /request-items/  {request\_id}/  {service\_id}/  delete/ | Удаляет услугу из заявки | request\_id: int,  service\_id: int | {  "message": "Service {service\_id} removed from request {request\_id}"  } |
| 4.1.15 | PUT | /request-items/  {request\_id}/  {service\_id}/  update/ | Обновляет данные по М-М (поле comment). | request\_id: int,  service\_id: int,  {  "comment": string  } | {  "request\_id": int,  "service\_id": int,  "message": "Comment updated",  "comment": string  } |
| 4.1.16 | POST | /user / | Регистрирует пользователя, доступно всем | {  username: string,  password: string,  is\_staff: bool,  is\_superuser: bool } | {  status : string } |
| 4.1.17 | PUT | /users/update/ | Обновляет данные пользователя. Доступно пользователю. | {  "username": string,  "password": string,  "email": string  } | {  "message": string,  "username": string,  "email": string } |
| 4.1.18 | POST | /users/login/ | Аутентификация пользователя. Доступно всем | {  "username": string,  "password": string  } | {  status: string  } |
| 4.1.19 | POST | /users/logout/ | Деавторизация, доступна дюбому аутентифицированному пользователю |  | {  status: string  } |