Введение

В рамках научно-проектной деятельности была продолжена работа над проектом разработки собственной web-платформы с интегрированной средой разработки (web-IDE) для нужд СКБ «ИРС». Предыдущие этапы включали в себя такие аспекты, как: создание прототипа превью-страницы; разработка тестового web-IDE; составление архитектуры конечной версии среды разработки.

На текущем этапе работа была сосредоточена над расширением функционала платформы и улучшением интерфейса, а также над добавлением новых компонентов, включая системы взаимодействия между пользователями.

1. ЦЕЛЬ ПРОЕКТА

Целью проекта является доработка и расширение web-платформы, сочетающей в себе функционал для написания и исполнения программного кода, а также реализацию механизмов взаимодействия между пользователями.

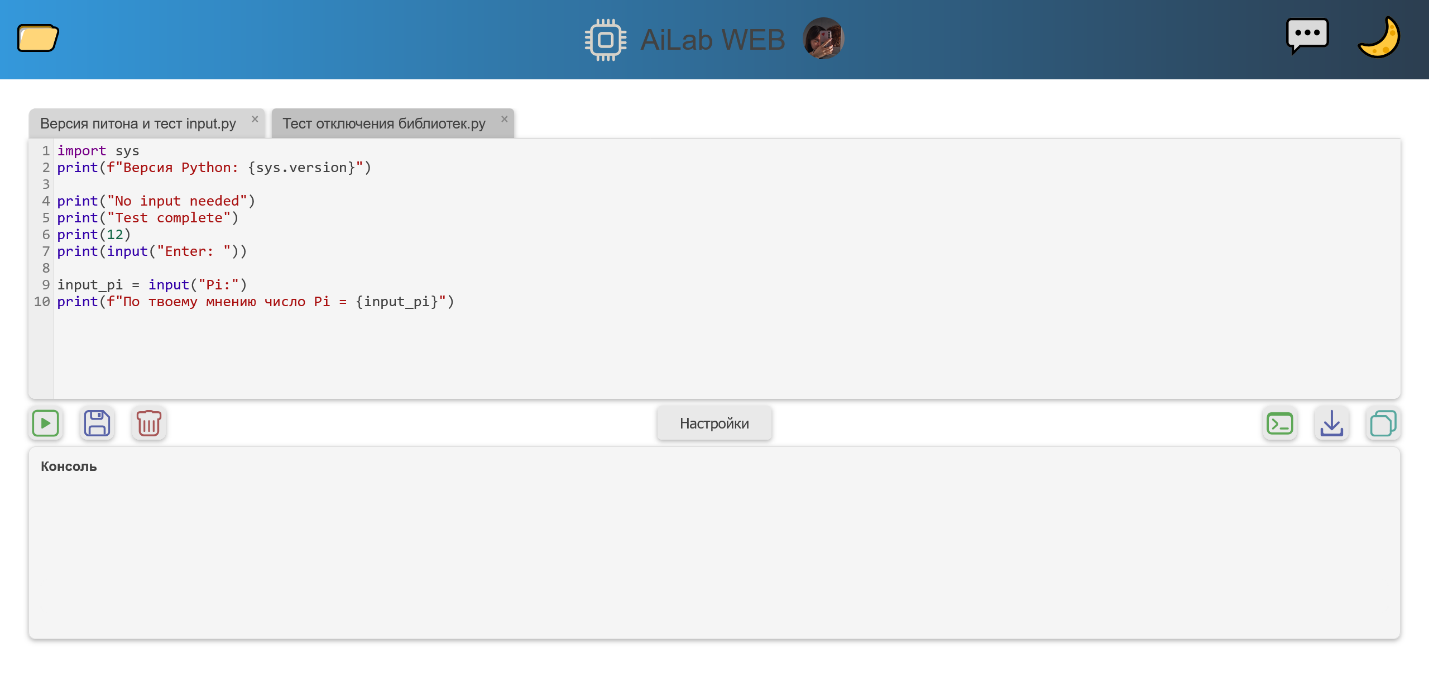
В рамках поставленной цели планируется такие действия, как:

* Переработка и улучшение интерфейса и логики работы web-IDE;
* Реализация системы регистрации, авторизации и управления профилями пользователей;
* Создание системы заявок в друзья и отображения списка контактов;
* Разработка мессенджера;
* Внедрение Искусственного Интеллекта (далее по тексту: ИИ) в мессенджер в виде чат-бота;
* Внедрение файлового менеджера;
* Повышение удобства и гибкости платформы для дальнейшего масштабирования.

1. ВЫПОЛНЕНИЕ ПОСТАВЛЕННЫХ ЗАДАЧ
   1. Переработка и улучшение интерфейса и логики работы web-IDE

В рамках данного этапа была выполнена переработка интерфейса и логики работы IDE-страницы. Старый интерфейс обладал рядом недостатков: он был неудобен для разработчика и ограничивал функциональные возможности платформы. В связи с этим было принято решение об обновлении как внешнего вида, так и внутренней логики взаимодействия элементов интерфейса.

Дополнительно была проведена визуальная переработка интерфейса IDE с целью повышения удобства и улучшения пользовательского восприятия. Изменения коснулись цветовой схемы, размещения элементов и логики взаимодействия элементов.

Рисунок 2.1 – Вид web-IDE.

В процессе доработки была удалена кнопка загрузки файла, как избыточная, и добавлена новая кнопка для открытия и закрытия консоли. Также внедрено меню управления средой выполнения кода.

Через это меню пользователь может:

* Изменять версию языка программирования;
* Подключать или отключать используемые библиотеки;
* В будущем — выбирать язык программирования.

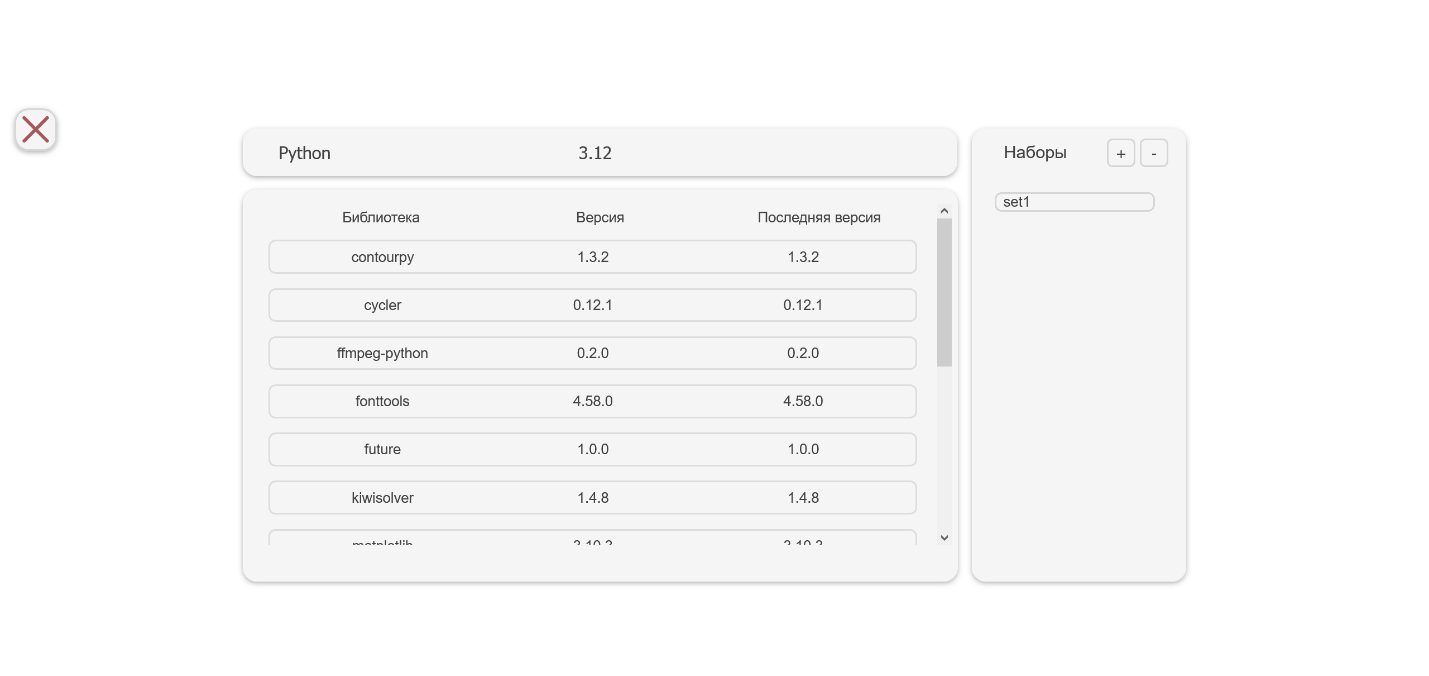
Для вызова данного меню была добавлена отдельная кнопка в панель управления.

Рисунок 2.2 – Вид меню управления средой выполнения кода.

* 1. Реализация системы регистрации, авторизации и управления профилями пользователей.

Одним из ключевых этапов стало создание системы работы с пользователями, включающая регистрацию, авторизацию и управление профилем. Это необходимо для индивидуализации работы в среде разработки, обеспечения безопасности и организации взаимодействия между участниками платформы.

Была реализована форма регистрации с базовой проверкой введённых данных, исключающей возможность повторной регистрации одного и того же пользователя. Также добавлена форма входа с переходом в личный кабинет после успешной авторизации.

После входа в систему пользователю становится доступен личный профиль, содержащий основную информацию: имя, адрес электронной почты, список друзей и QR-код, содержащий ссылку на страницу пользователя. Всю эту информацию пользователь лично может изменить в настройках профиля.

Через страницу пользователя можно открыть список друзей и добавить новые контакты.

При необходимости пользователь может изменить часть данных, или добавить дополнительную информацию, например: должность; номер телефона; ссылки на такие ресурсы, как «Telegram», «VK» и «GitHub». Также предусмотрен механизм отправки заявок на добавление в друзья и редактирования профиля в виде меню, где можно изменить данные и фото профиля.

Для хранения данных о пользователях используется разработанная база данных SQLite [1]. Логика работы реализована на стороне сервера.

Реализация этих функций позволила заложить основу для организации взаимодействия между пользователями и для последующего расширения платформы.

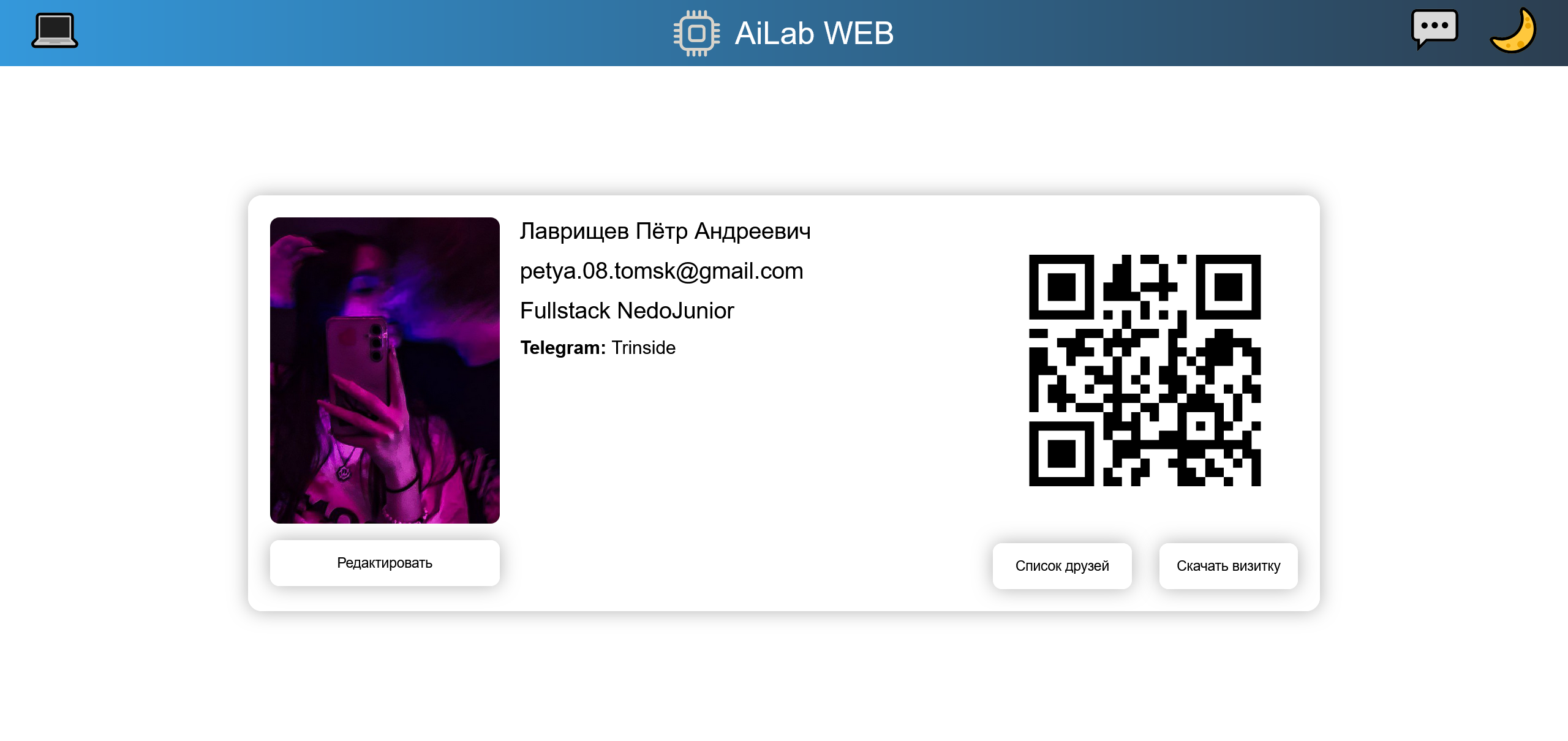
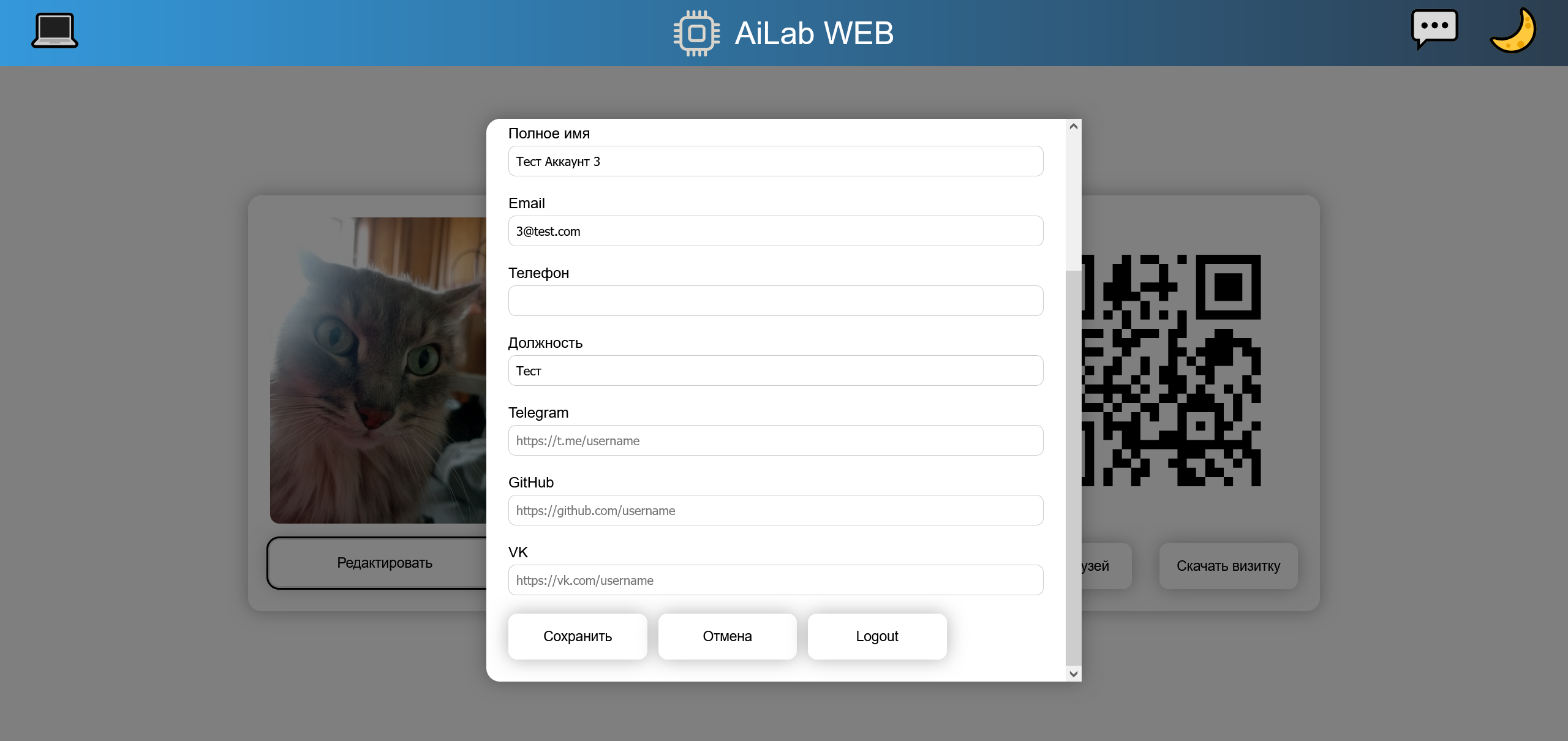
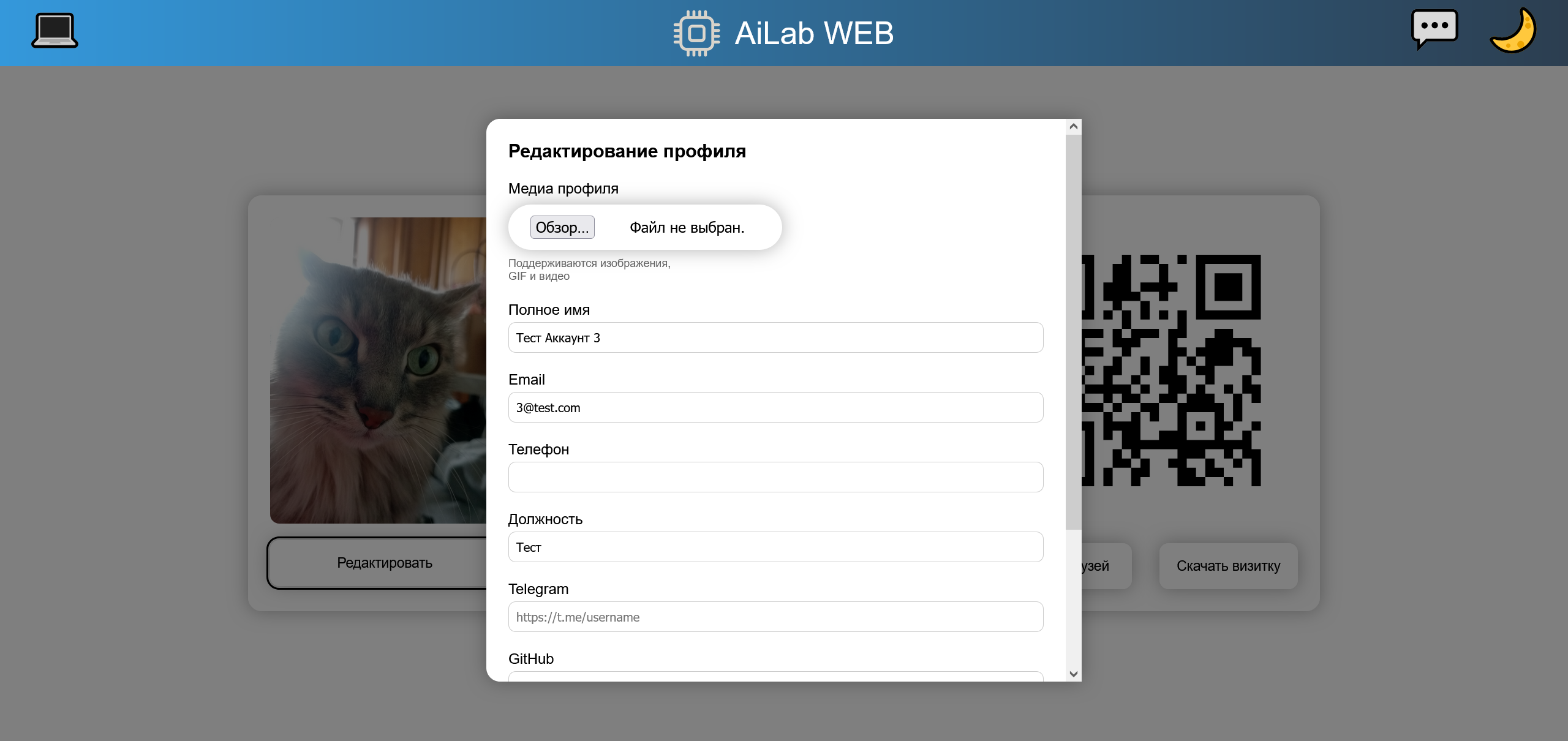


Рисунок 2.3 – Вид профиля пользователя.



а) б)

Рисунки 2.4 (а-б) – Вид меню редактирования профиля

* 1. Создание системы заявок в друзья и отображения списка контактов

Для повышения уровня взаимодействия между пользователями платформы была разработана система заявок в друзья, а также реализовано отображение списка контактов.

Пользователь имеет возможность найти другого участника платформы и отправить ему заявку в друзья. Отправленный запрос отображается у получателя, который может либо принять, либо отклонить его. При принятии заявки оба пользователя автоматически добавляются в списки контактов друг друга.

Контакты отображаются в виде отдельного списка, доступного из пользовательского профиля. В дальнейшем именно с этими пользователями можно будет начинать диалог в мессенджере.

Данная функция дает основу для реализации социальной составляющей web-платформы, то есть обеспечивает взаимодействие между участниками.

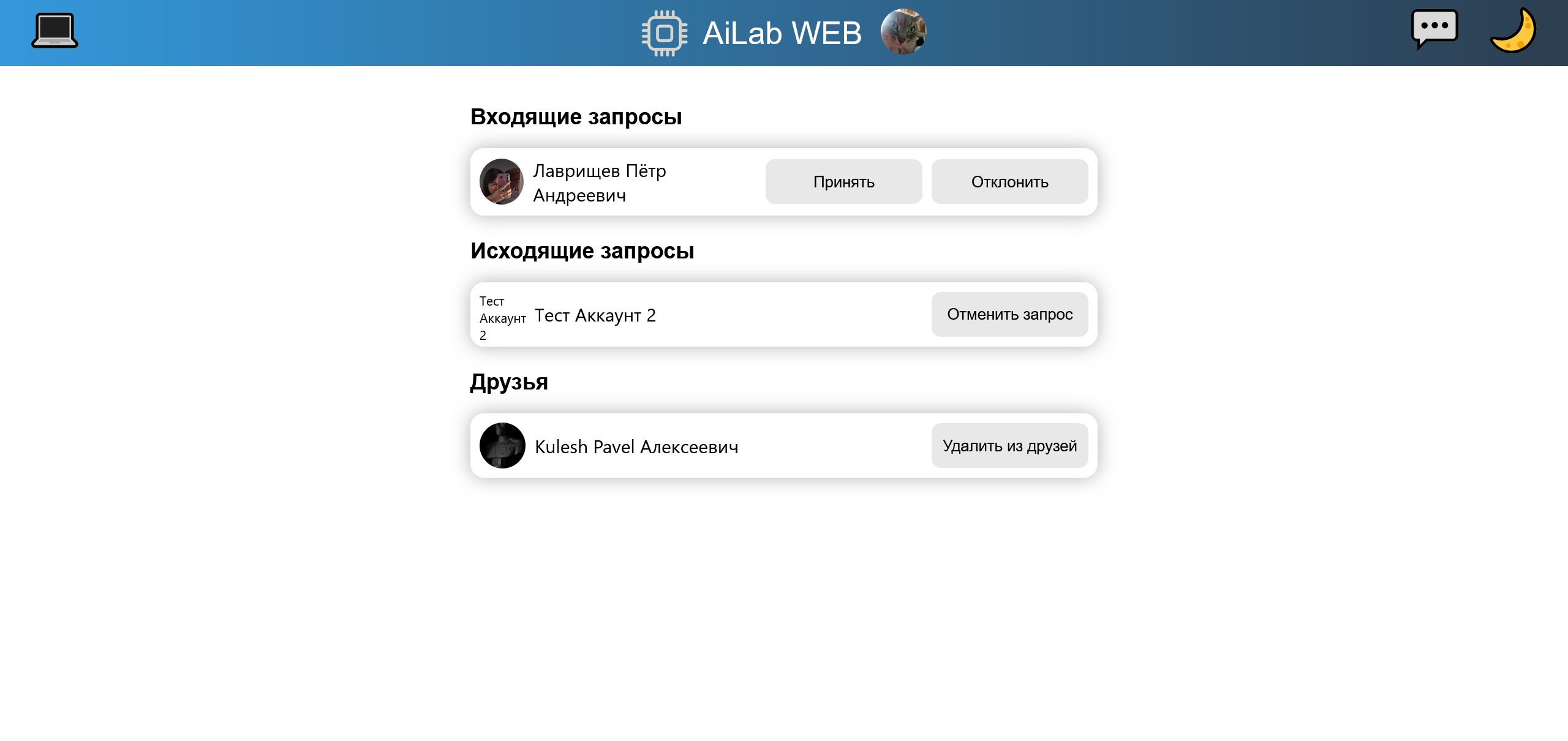


Рисунок 2.5 – Вид страницы списка друзей.

* 1. Разработка мессенджера

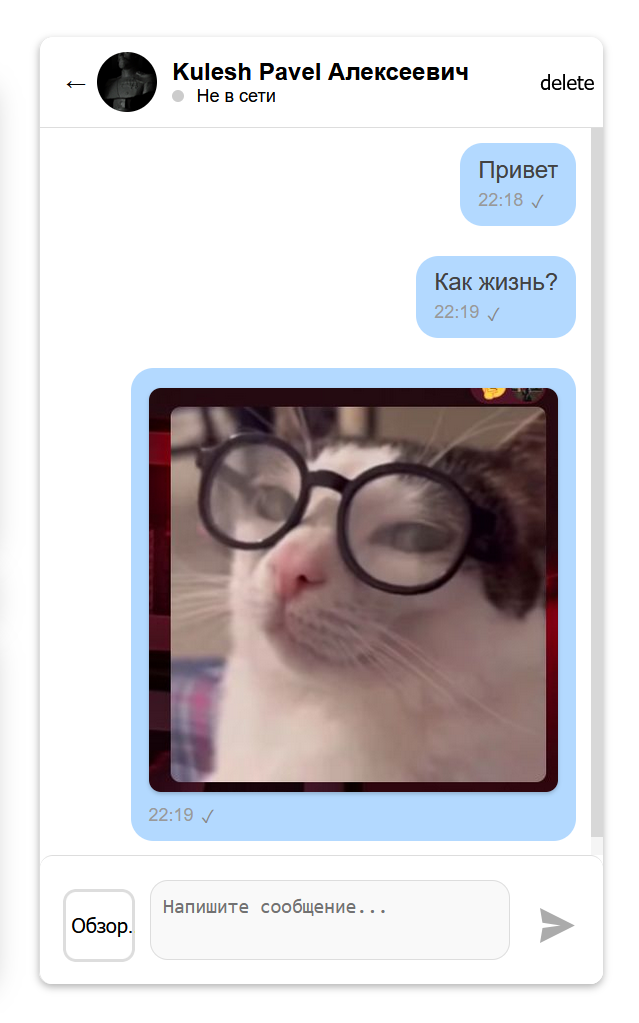
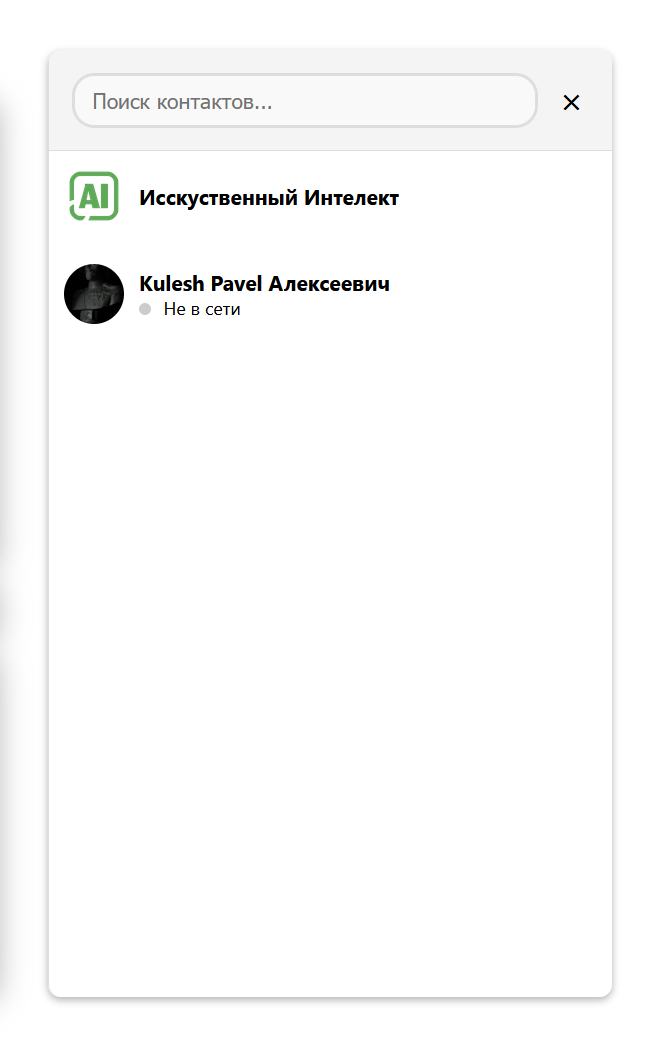
Следующим этапом работы стала реализация встроенного мессенджера, предназначенного для обмена сообщениями между пользователями платформы в режиме реального времени.

Мессенджер позволяет пользователю:

* Вести личную переписку с добавленными в друзья контактами;
* Просматривать историю сообщений;
* Общаться с ИИ в чате;

Интерфейс мессенджера разработан с упором на простоту и удобство: каждое сообщение визуально разделено, отображается имя отправителя, время отправки и статус просмотра. Можно отправлять как текстовые сообщения, так и фото, видео и файлы. Также можно удалить историю переписки.

Создание мессенджера значительно повысило уровень интерактивности платформы и позволило пользователям оперативно обмениваться информацией, не покидая среду разработки. Это также стало важным подготовительным шагом к интеграции дополнительных интеллектуальных функций, таких как чаты с ИИ.



а) б)

Рисунки 2.6 (а-б) – Список контактов и переписка в мессенджере.

* 1. Внедрение ИИ в мессенджер в виде чат-бота

Для расширения функциональных возможностей платформы и повышения её практической ценности в учебной и исследовательской деятельности, в мессенджер был внедрён чат-бот на основе искусственного интеллекта.

Пользователь может перейти в диалог с виртуальным собеседником перейдя на страницу списка чатов с ИИ, после чего можно создать новый чат и перейти в него, либо перейдя в уже существующий чат.

Интерфейс чата с ИИ не отличается от обычного диалога, что обеспечивает единообразие системы чатов.

При написании ИИ использовались современные библиотеки для работы с большими языковыми моделями (Large Language Models LLM), обработки документов и поиска в интернете [2].

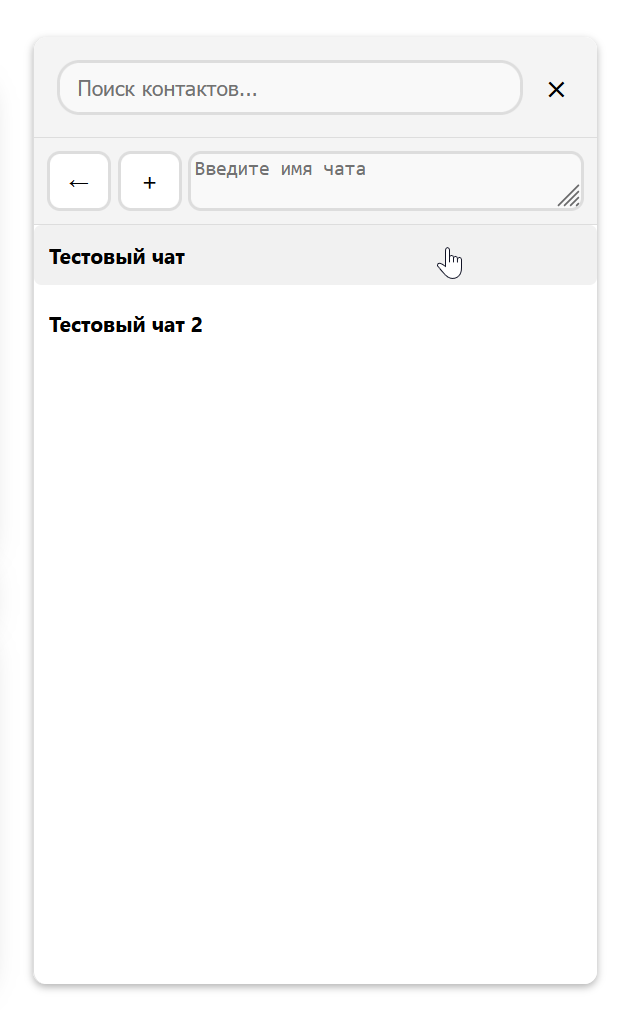
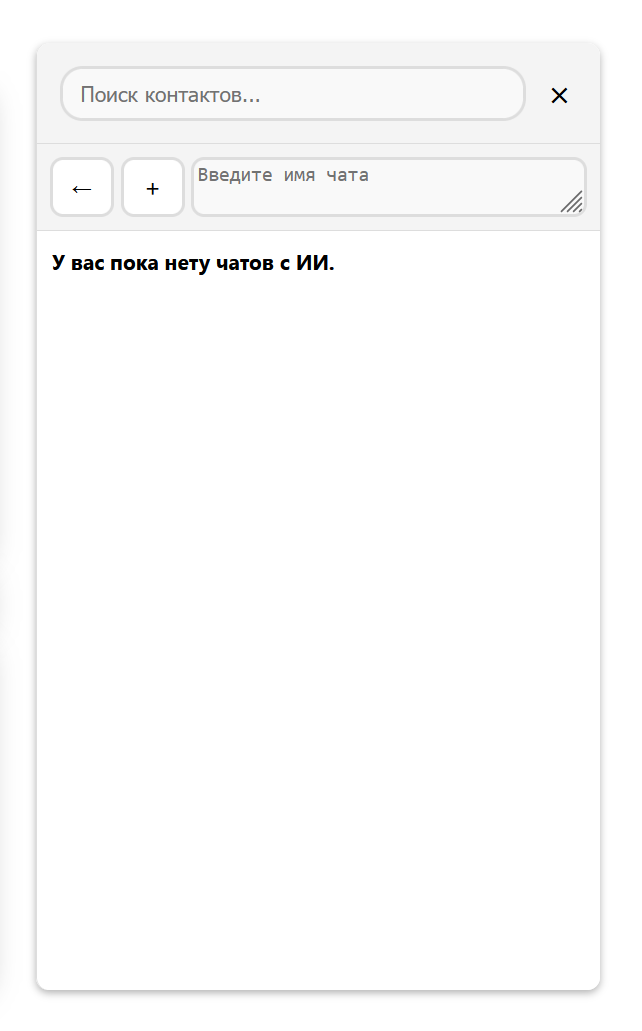
Нейронная сеть предоставляет возможности:

1. Обрабатывать пользовательские запросы с учетом контекста из файлов и истории диалога.
2. Загружать содержимое различных типов файлов (PDF, DOCX, TXT, CSV и др.).
3. Сохранять историю общения в файл для поддержания контекста.
4. Использовать языковую модель для генерации ответов.

В коде используется языковая модель Mistral, интегрированная через OllamaLLM. Mistral — это современная модель для обработки естественного языка, оптимизированная для задач генерации текста, ответа на вопросы и других NLP-задач [3]. Она работает через платформу Ollama, которая позволяет запускать LLM локально или на сервере [3]. Код не предоставляет деталей о версии или конфигурации модели Mistral, но предполагается, что она настроена для эффективной работы с текстовыми запросами.

Добавление чат-бота позволило не только повысить интерактивность платформы, но и предоставило пользователям возможность оперативно получать помощь и сопровождение в процессе разработки.

На рисунках 2.7 - 2.8 видна работа с ИИ и чатами:



а) б)

Рисунки 2.7 (а-б) – Список чатов с ИИ, и создание новых.

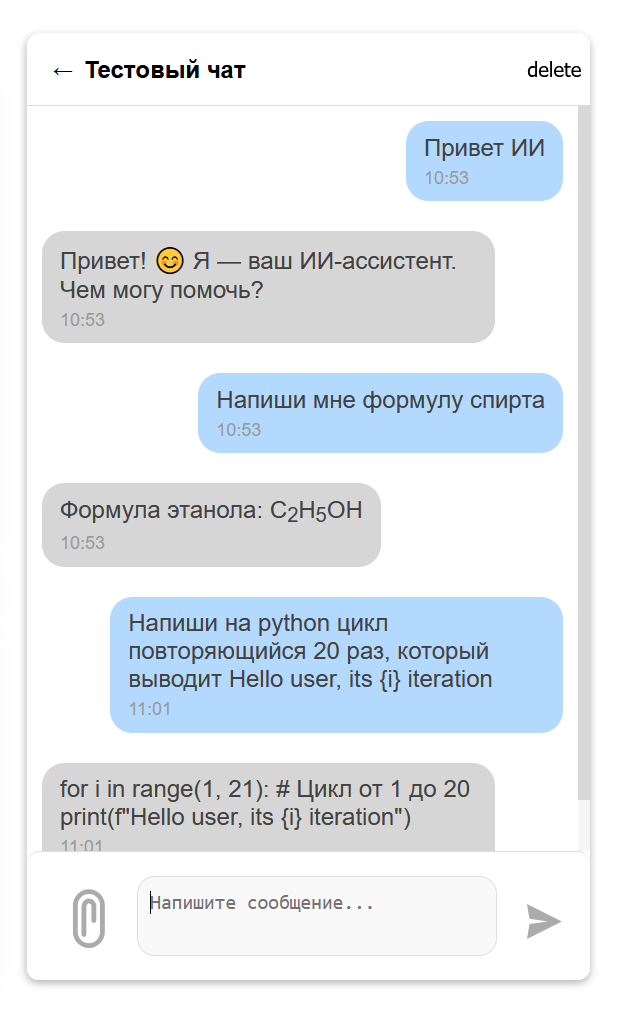


Рисунок 2.8 – Чат с ИИ.

* 1. Внедрение файлового менеджера

В рамках расширения функциональности платформы был разработан встроенный файловый менеджер, обеспечивающий пользователю базовые возможности по управлению своими файлами внутри web-IDE.

Файловый менеджер предоставляет такие функции как:

* Просмотр файлов пользователя.
* Создание новых файлов.
* Переименование и удаление файлов.
* Открытие файлов для редактирования и запуска в IDE.

Интерфейс файлового менеджера размещён сбоку от основного рабочего пространства и выполнен в минималистичном стиле, обеспечивающем удобный и быстрый доступ к необходимым материалам. Работа с файлами реализована таким образом, чтобы исключить доступ к чужим данным и обеспечить базовый уровень безопасности.

Добавление данной функции позволило сделать работу в IDE более автономной и приближённой к привычным средам разработки, а также подготовило основу для возможной интеграции с системами хранения и синхронизации файлов.

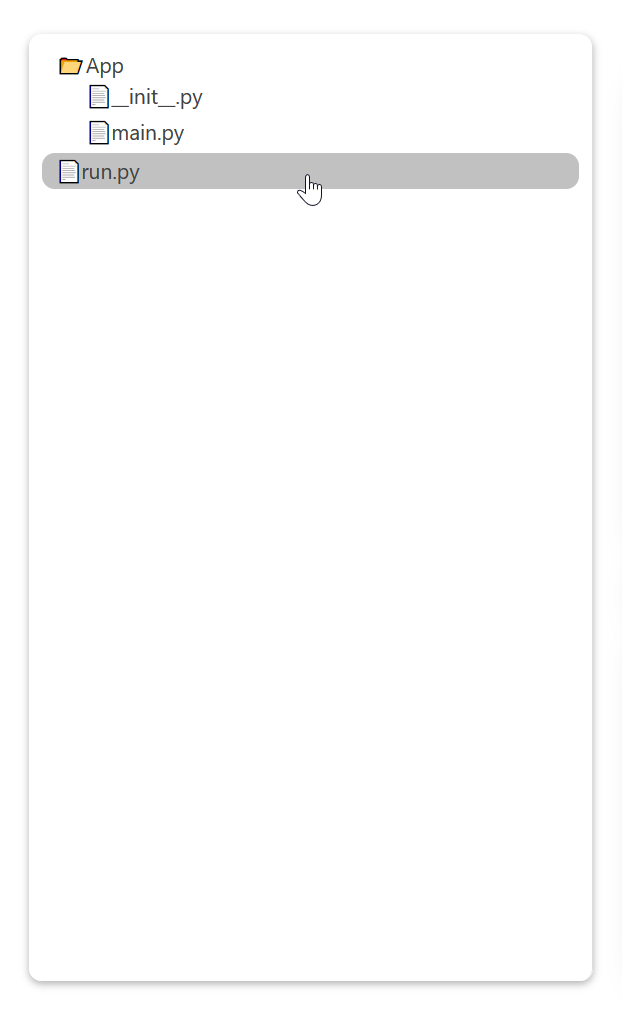


Рисунок 2.9 – Файловый менеджер.

* 1. Повышение удобства и гибкости платформы для дальнейшего масштабирования

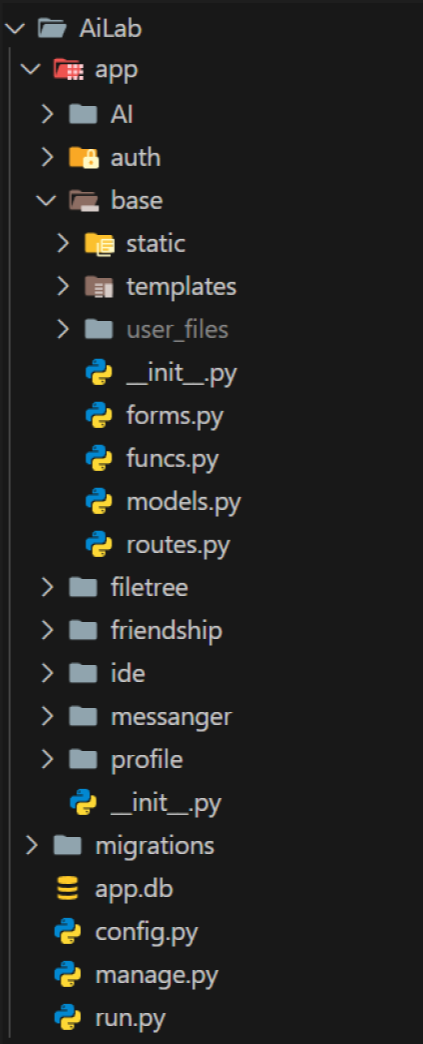
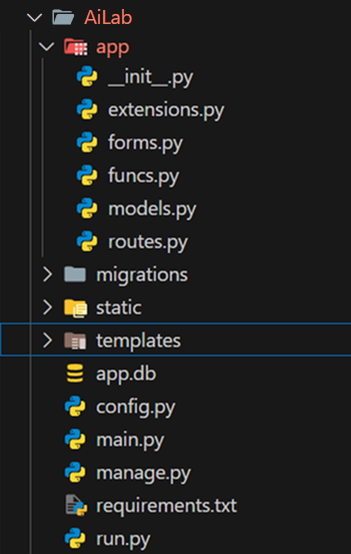
Для повышения стабильности и удобства работы с платформой, а также подготовки её к дальнейшему расширению, была проведена серия доработок в структуре проекта и пользовательском интерфейсе.

В рамках данного этапа:

* Переработаны отдельные элементы дизайна для улучшения визуального восприятия и повышения удобства навигации;
* Устранён ряд багов, связанных с отображением интерфейсных компонентов, сохранением кода и переключением между вкладками;
* Улучшена структуризация проекта: разделены модули по функциональности (авторизация, IDE, мессенджер, профили, файловый менеджер, логика пользователей), упорядочены статические ресурсы и шаблоны;
* Упрощена логика маршрутизации на серверной стороне.

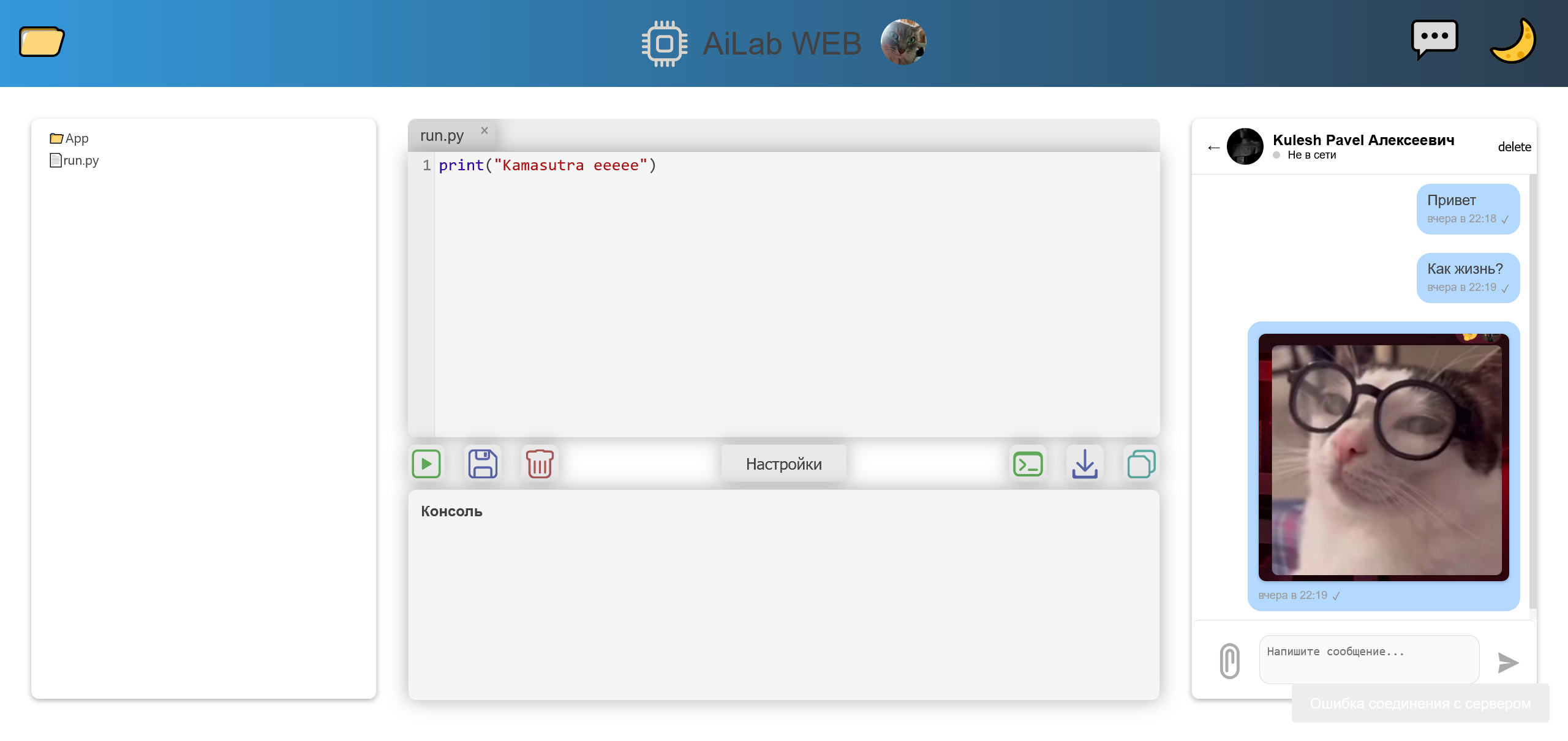
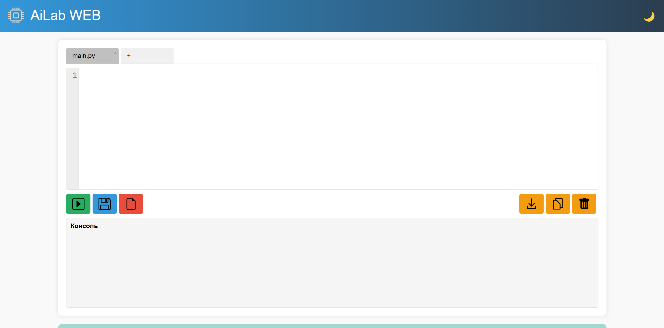
Эти изменения сделали проект более устойчивым, упростили его поддержку и создали задел для внедрения новых функций в будущем — без необходимости переработки уже существующего кода.

На рисунках ниже показаны изменения:



а) б)

Рисунки 2.10 (а-б) – Файлы до и после реструктуризации.



а) б)

Рисунки (а-б) 2.11 – Изменения комплексного вида платформы.

1. ТЕХНИЧЕСКАЯ РЕАЛИЗАЦИЯ
   1. Технологический стек

Разработка велась с применением следующих технологий:

* HTML\CSS – разметка и стилизация интерфейса [5];
* JavaScript – реализация интерактивных элементов, обмен данными с сервером, запросы на сервер, обработка клиентской логики [5];
* Flask – фреймворк для создания серверного приложения и организации маршрутов [6];
* Flask-SockeIO – обмен данными между клиентом и сервером в реальном времени [7];
* Eventlet – библиотека для асинхронной обработки событий, используемая в связке с Flask-SocketIO;
* SQLite – база данных для хранения информации о пользователях, друзьях, сообщениях, чатах и файлах;
* CodeMirror – редактор кода с встроенной подсветкой синтаксиса [8];
* Python – серверная часть, обработка серверной логики [10];
* Flask-WTF – создание и обработка web-форм с встроенной валидацией и защитой от «CSRF»-атак [11];
* Flask-Login – управление авторизацией, сессиями и доступом к закрытым разделам платформы [12];
* QR Code – генерация QR-кодов для профилей пользователей [13];
* OllamaLLM – библиотека для удобной интеграции языковых моделей Ollama в Python-приложения;
* Ollama – платформа для локального и серверного запуска больших языковых моделей (LLM);
* Mistral – современная языковая модель для генерации текста и обработки естественного языка, работающая через Ollama.
  1. Структура проекта

Проект разделен на модули, что упрощает поддержку:

* auth/ – регистрация и авторизация пользователей;
* ide/ – логика работы среды разработки;
* messenger/ – система обмена сообщениями в мессенджере;
* profile/ – управление профилями пользователей;
* filetree/ – файловый менеджер;
* friendship/ – логика работы системы добавления в друзья;
* AI/ – обработка сообщений пользователей ИИ;
* base/ – основной модуль содержащий:
  + модели базы данных;
  + формы на основе «Flask-WTF»;
  + обработка подключений сессий по «SocketIO»;
  + маршрут для превью-страницы платформы.
* static/ – статичные файлы: CSS стили, JavaScript код, изображения;
* templates/ – HTML шаблоны, используемые Flask;
* user\_files/ – каталог с индивидуальными файлами пользователей:
  + загруженные и созданные файлы;
  + пресеты среды выполнения (язык, версия, библиотеки);
  + контексты для чатов с ИИ.
  1. Хранение и обработка файлов

Все данные о пользователях, сообщениях, друзьях и файлах хранятся в базе данных SQLite, доступ к которой осуществляется через «ORM» или «SQL»-запросы. Пароли сохраняются в зашифрованном виде. Для каждой сущности реализованы отдельные модели, обеспечивающие модульность и безопасность обработки. Файлы пользователей сохраняются в уникальные каталоги, изолированные друг от друга.

* 1. Безопасность

Особое внимание в разработке платформы уделялось обеспечению безопасности пользовательских данных и защите от типовых уязвимостей. Все формы регистрации, авторизации и отправки данных на сервер сопровождаются обязательной валидацией как на стороне клиента, так и на стороне сервера. Это позволяет предотвратить некорректные или вредоносные запросы.

Для защиты от «**SQL**»**-инъекций** используется параметризация запросов, что исключает возможность внедрения вредоносного кода через поля ввода. В обработке пользовательских сообщений и вводимых данных реализована очистка и экранирование «HTML»-содержимого, что предотвращает «**XSS**»**-атаки** (межсайтовый скриптинг) — попытки внедрить и выполнить сторонний JavaScript-код **[9].**

Также предусмотрены базовые меры защиты от **CSRF-атак** (межсайтовой подделки запроса) за счёт ограничения доступа к функциям только для авторизованных пользователей и использования безопасной обработки форм.

Для всех защищённых маршрутов реализована проверка прав доступа и статуса аутентификации, а попытки несанкционированного доступа перенаправляются на страницу входа. Кроме того, структура URL и логика маршрутизации устроены таким образом, чтобы исключить прямой доступ к критически важным элементам сервера или чужим данным.

В совокупности эти меры обеспечивают устойчивую работу платформы к большинству распространённых угроз.

1. ИТОГИ

В ходе научно-проектной деятельности была существенно расширена функциональность web-платформы для СКБ «ИРС». Работа охватывала как технические аспекты, так и вопросы взаимодействия между пользователями.

Выполнены все поставленные задачи, в результате чего платформа значительно развилась: из базового прототипа она превратилась в полноценную web-среду, объединяющую инструменты для разработки программного кода, взаимодействия между пользователями и управления пользовательскими данными.

Система стала удобной, расширяемой и безопасной, а её архитектура теперь позволяет в дальнейшем быстро внедрять новые функции и улучшать уже существующие модули

Заключение

В результате проведённой работы была создана стабильная, функциональная и визуально доработанная версия web-платформы для СКБ «ИРС». Платформа объединила в себе ключевые элементы, необходимые для комфортной индивидуальной и командной разработки: web-IDE, систему пользователей, мессенджер, чаты с ИИ и файловый менеджер.

В ходе реализации были улучшены как интерфейсные, так и внутренние структурные компоненты проекта, обеспечена базовая защита от распространённых атак и подготовлена архитектура для будущего расширения.

Развитие проекта будет направлено на дальнейшее повышение удобства и эффективности работы пользователей, а также на постепенное расширение функциональности.

Список используемых источников

1. SQLite [Электронный ресурс]: документация SQLite URL: https://www.sqlite.org/docs.html (Дата обращения: 25.03.2025);
2. Large Language Models LLM [Электронный ресурс]: статья о LLM URL: https://en.wikipedia.org/wiki/Large\_language\_model (Дата обращения: 12.05.2025);
3. Mistral [Электронный ресурс]: сайт Mistral URL: https://mistral.ai/ (Дата обращения: 25.03.2025);
4. Ollama [Электронный ресурс]: сайт Ollama URL: https://ollama.com/ (Дата обращения: 25.03.2025);
5. HTML, CSS, JS [Электронный ресурс]: Документация MDN Web Docs https://developer.mozilla.org/ru/docs/Web/HTML (Дата обращения: 26.02.2025);
6. Flask [Электронный ресурс]: документация библиотеки Flask URL: https://flask.palletsprojects.com/en/stable/ (Дата обращения: 12.03.2025);
7. Flask-SocketIO [Электронный ресурс]: документация Flask-SocketIO URL: https://flask-socketio.readthedocs.io/en/latest/ (Дата обращения: 16.03.2025);
8. CodeMirror [Электронный ресурс]: сайт Code Mirror URL: https://codemirror.net/ (Дата обращения: 06.03.2025);
9. XSS [Электронный ресурс]: Статья Skillfactory: Что такое XSS URL: https://blog.skillfactory.ru/glossary/xss/ (Дата обращения: 15.05.2025);
10. Python [Электронный ресурс]: сайт python URL: https://python.org (Дата обращения: 10.05.2025);
11. Flask-WTF [Электронный ресурс]: Документация Flask-WTF URL: https://flask-wtf.readthedocs.io/en/1.2.x/ (Дата обращения: 20.05.2025);
12. Flask-Login [Электронный ресурс]: Документация Flask-Login  URL: https://flask-login.readthedocs.io/en/latest/ (Дата обращения: 15.05.2025);
13. QR Code [PIL] [Электронный ресурс]: Сайт библиотеки qrcode URL: https://pypi.org/project/qrcode/ (Дата обращения: 15.05.2025);