

1 Постановка задачи

Для достижения цели данной работы были поставлены следующие задачи:

- изучение особенностей различных браузеров и их поддержки технологий;
- разработка дизайна и интерфейса приложения;
- создание базы данных для хранения информации о задачах;
- разработка функционала для добавления, удаления и редактирования задач;
- разработка дополнительных виджетов, например, трекер целей и заметки.

Идея создания приложений для планирования дел возникла в результате растущей потребности людей в эффективном управлении своим временем и задачами. С развитием технологий и внедрением смартфонов в повседневную жизнь, стало возможным создание приложений, которые помогают людям организовать свой рабочий и личный график, следить за выполнением задач и не забывать о важных событиях.

Первые приложения для планирования дел начали появляться в начале 2000-х годов, когда мобильные устройства стали доступны для широкой аудитории. Они предоставляли возможность создавать список задач и напоминать о них в нужное время. Однако, эти приложения были довольно простыми и не предлагали широкий функционал.

С появлением смартфонов и распространением мобильного интернета, стали появляться более продвинутые приложения для планирования дел. Они предоставляют возможность создавать подробные задачи с описанием, прикреплять файлы и ссылки, делиться списками задач с другими пользователями, планировать задачи на конкретную дату и время, а также синхронизировать данные между устройствами.

Сегодня, приложения для планирования дел стали неотъемлемой частью повседневной жизни многих людей. Они помогают организовать

рабочий и личный график, увеличить производительность и сократить время, затрачиваемое на рутинные задачи.

Создание приложения для планирования дел является актуальным в настоящее время по нескольким причинам:

- увеличение производительности. В современном мире у людей всё больше дел и задач, и им нужно находить эффективные способы управления своим временем. Приложение для планирования дел может помочь увеличить производительность, оптимизировать расписание и улучшить управление временем;
- уменьшение стресса. Отсутствие планирования может приводить к чувству беспокойства и стресса, так как человек может забывать о важных делах или не успевать их выполнить. Приложение для планирования дел поможет избежать таких ситуаций и снизить стресс;
- улучшение организации. Приложение для планирования дел позволяет лучше организовать свою жизнь и повысить уровень самодисциплины. Это может привести к более успешному достижению целей и улучшению качества жизни;
- доступность. Приложение для планирования дел доступно на любом устройстве с интернет-соединением, что позволяет использовать его в любой момент и в любом месте. Это удобно для людей, которые постоянно находятся в движении;
- совместное использование. Приложение для планирования дел может быть использовано одновременно несколькими людьми, что позволяет организовать работу в коллективе и совместно работать над выполнением задач.

2 Проектирование базы данных

На этом этапе создается структура базы данных, которая будет использоваться в приложении. В данном случае используется база данных SQLite [1], поэтому необходимо определить, какие таблицы и поля будут в базе данных.

Registration customuser — главная таблица базы данных проекта. Является расширенной версией таблицы User, предоставленной фреймворком изначально. Содержит в себе такие поля как:

- id — идентификатор строки в таблице;
- password — хэш и метаданные о пароле;
- last login — дата и время последнего входа пользователя в систему;
- is superuser — логический. Обозначает, что у этого пользователя есть все разрешения без их явного назначения;
- username — никнейм пользователя;
- first name — имя пользователя;
- last name — фамилия пользователя;
- is staff — логический. Определяет, может ли этот пользователь получить доступ к сайту администратора;
- is active — логический. Определяет, может ли этот пользователь получить доступ к сайту администратора;
- date joined — дата и время, обозначающие, когда была создана учетная запись. Устанавливается на текущую дату/время по умолчанию при создании учетной записи;
- avatar — поле, хранящее путь файла аватара пользователя.

Todolist — таблица для хранения данных о деле, которое необходимо выполнить. Имеет связь «один ко многим», которая соединяет конкретного пользователя с несколькими делами. Содержит в себе такие поля как:

- todo — текстовое поле, содержащее описание дела;
- checked — логическое поле, хранящее данные о том, было ли дело выполнено;

- day — день, на который планируется дело;
- month — месяц, на который планируется дело;
- year — год, на который планируется дело;
- user id — id пользователя, с помощью которого осуществляется

«запоминание» к какому пользователю относится данная запись.

Имеет поле id, описанное в предыдущем пункте.

Notes — таблица заметок, использующая связь «один ко многим».

Содержит следующие поля:

- topic — заголовок заметки;
- text — основной текст заметки;
- num — уникальный идентификатор, позволяющий обратиться к

необходимой заметке без использования ее id.

Имеет поля id и user id, описанные в предыдущих пунктах.

Goals — таблица, хранящая цели пользователя. Соединена с таблицей registration customuser по раннее описанному принципу. Имеет следующие поля:

- goal — поле, содержащее текст цели.

Имеет поля id и user id, описанные в предыдущих пунктах.

Podgoals — таблица подцелей. Связана с таблицей registration customuser и goals. Содержит следующие поля:

- podgoal — основной текст подцели;
- checked — логическое поле, хранящее данные о том, было ли дело выполнено;
- index — индекс подцели для соответствующей цели;
- goal id — id цели, с помощью которого осуществляется

«запоминание» к какой цели относится данная запись.

Имеет поля id и user id, описанные в предыдущих пунктах.

3 Создание макета интерфейса

При проектировании веб-приложения WoMo были реализованы

следующие функциональные возможности приложения, которые доступны пользователю:

- регистрация в приложении;
- доступ к личному аккаунту в системе;
- планирование текущего календарного дня;
- возможность добавления дел на текущую неделю;
- календарь с функцией редактирования задач на месяц;
- трекер личных целей;
- добавление, редактирование и удаление заметок.

Подробная структура веб-приложения представлена на рисунке 1.



Рисунок 1 – Структура веб-приложения WoMo

В процессе проектирования были разработаны прототипы экранов приложения с помощью веб-сервиса Figma. Был создан черновой вариант макета, который помог определить расположение элементов интерфейса на экране и основные элементы управления.

При создании прототипов были разработаны мокапы, дающие представление о внешнем виде будущего веб-приложения.

Макет главной страницы приложения, где можно добавлять задачи на текущий день и перейти к остальным окнам, представлен на рисунке 2.

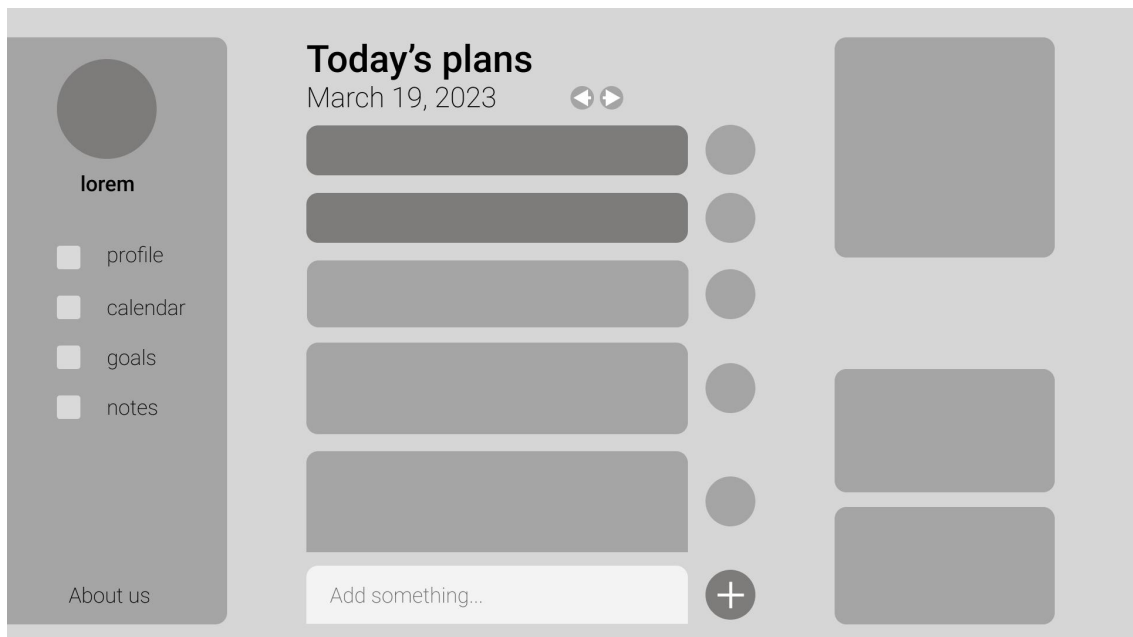


Рисунок 2 – Макет главной страницы приложения

На рисунке 3 продемонстрирован прототип окна текущей недели, где можно будет просмотреть задачи на каждый день в отдельности. Из данного окна с помощью нажатия на кнопку можно будет перейти на страницу с планером на месяц, макет которого представлен на рисунке 4.

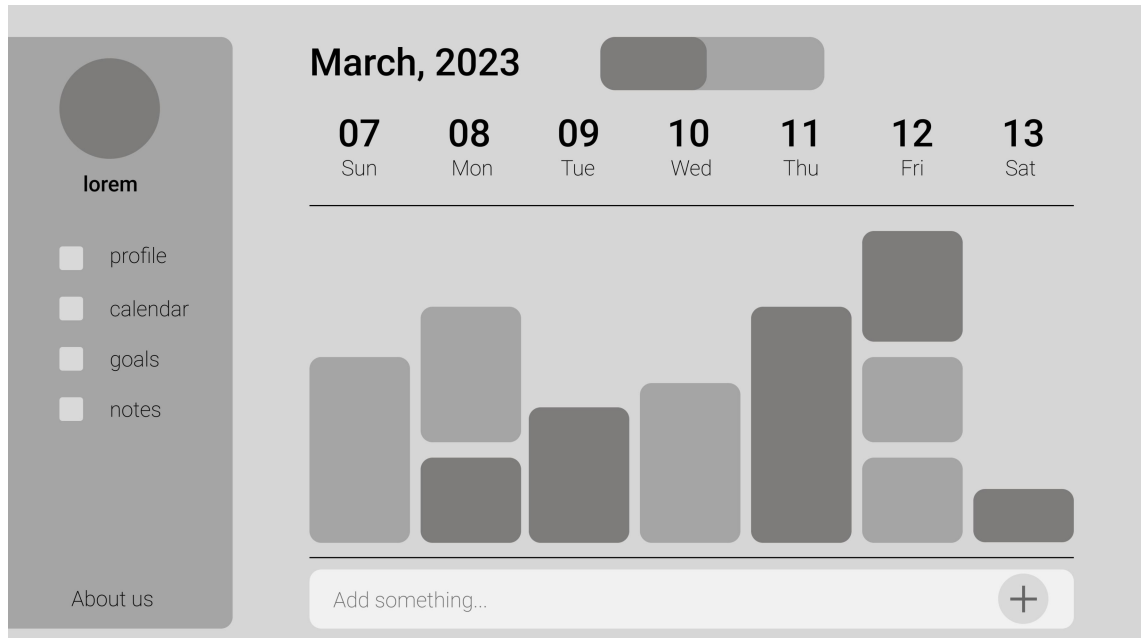


Рисунок 3 – Прототип страницы планирования недели



Рисунок 4 – Макет страницы с календарём на месяц

Также, в работе был реализован прототип трекера целей, который представлен на рисунке 5.

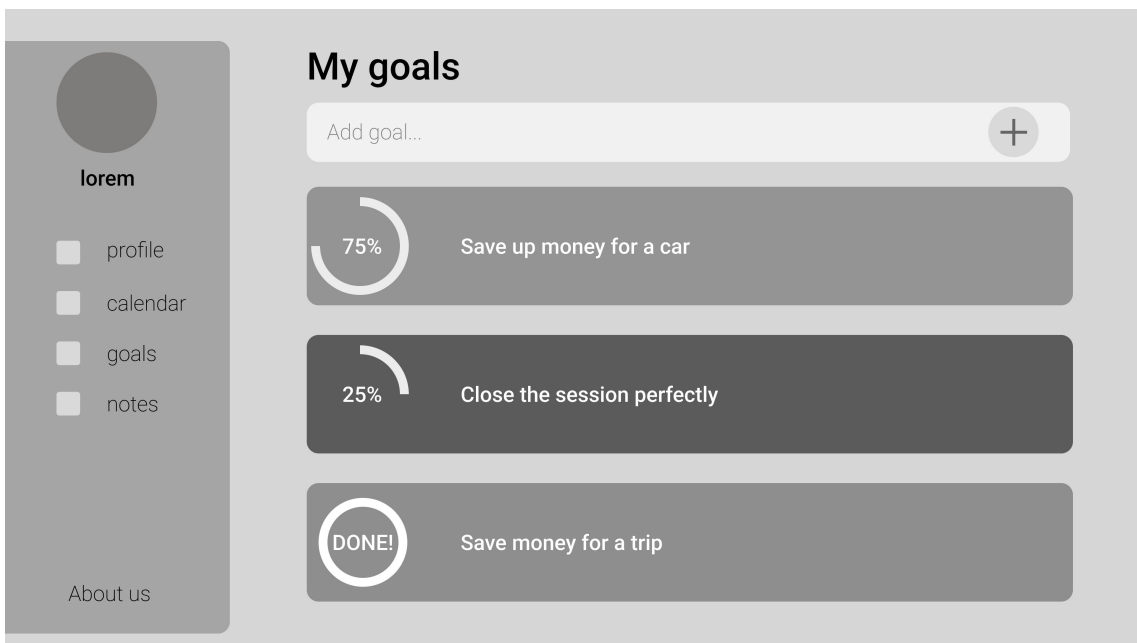


Рисунок 5 – Прототип страницы с трекером целей

Далее, было разработано схематичное изображение окна, где можно управлять личными заметками. Результат работы представлен на рисунке 6.

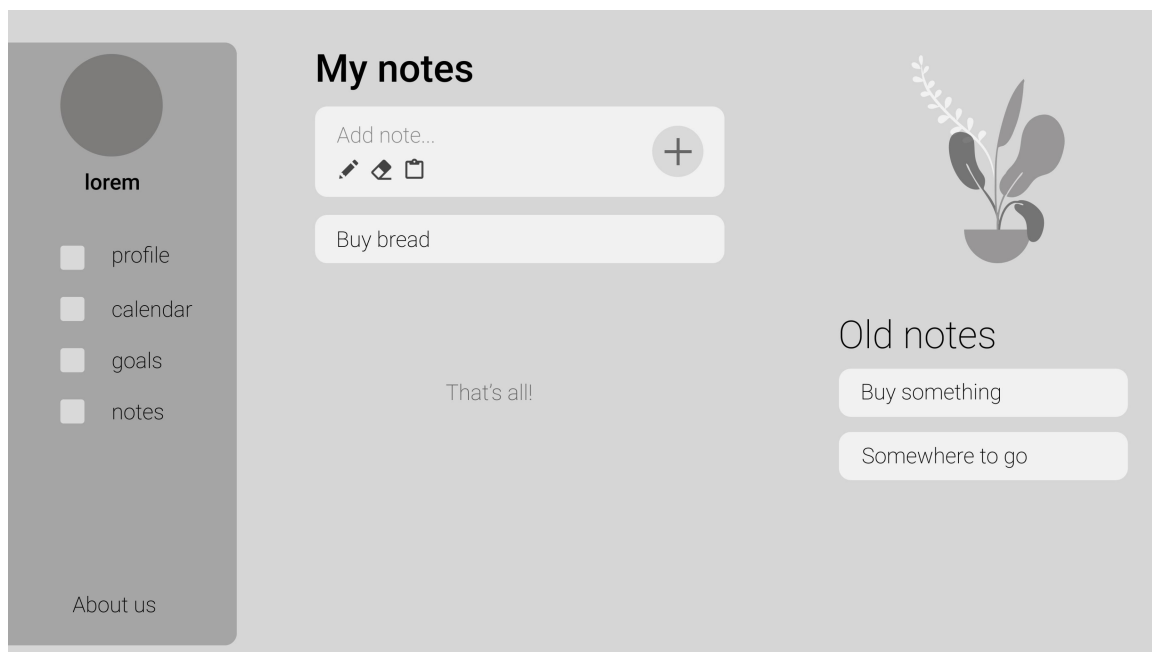


Рисунок 6 – Макет страницы управления личными заметками

На рисунке 7 изображен макет страницы личного кабинета пользователя веб-приложения.

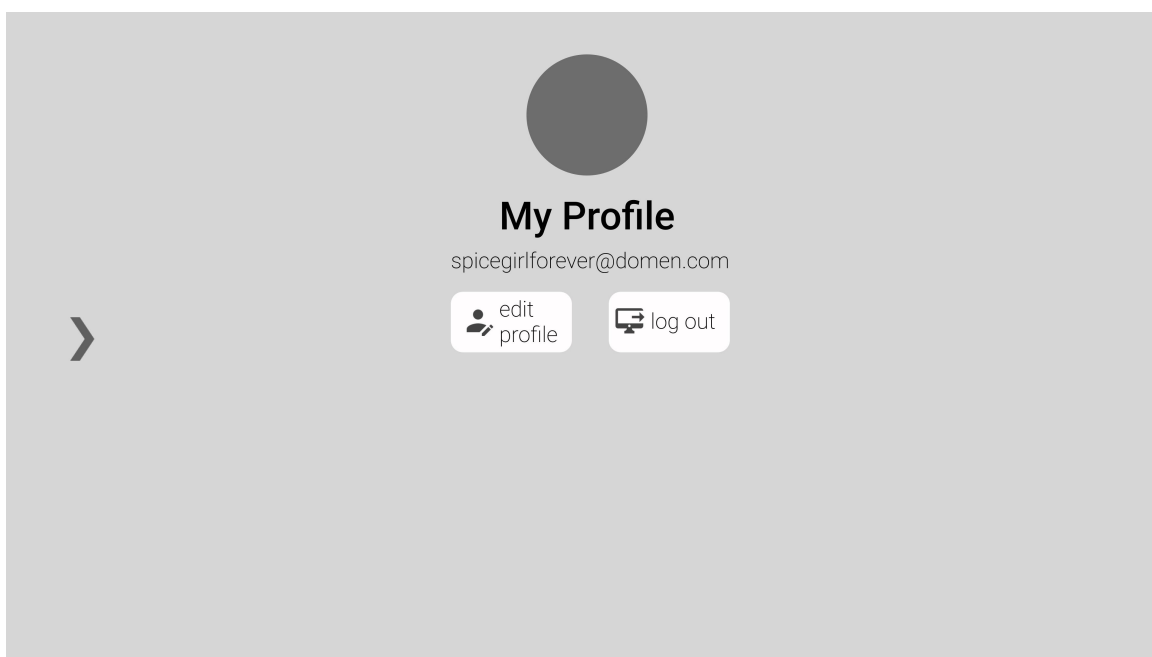


Рисунок 7 – Прототип страницы личного кабинета пользователя

В ходе выполнения проектирования работы была проведена оценка удобства использования приложения, исходя из эксперимента произведены следующие правки:

— на странице расписания на месяц нужно добавить вывод списка задач на необходимый день при нажатии на него;

— для корректного вычисления процента достижения локальной цели необходимо добавить на соответствующей странице всплывающее окно, позволяющее разбить цель на несколько менее объемных;

— в главном меню добавить ещё одну ссылку, переносящую на страницу с полезной информацией.

С учетом вышеперечисленных правок был разработан итоговый макет интерфейса, где были соблюдены все требования к функциональности и удобству использования.

При входе в приложение пользователь должен попасть на страницу входа, где предлагается войти либо в существующий аккаунт, либо зарегистрироваться. Результат разработки интерфейса страницы представлен на рисунке 8.

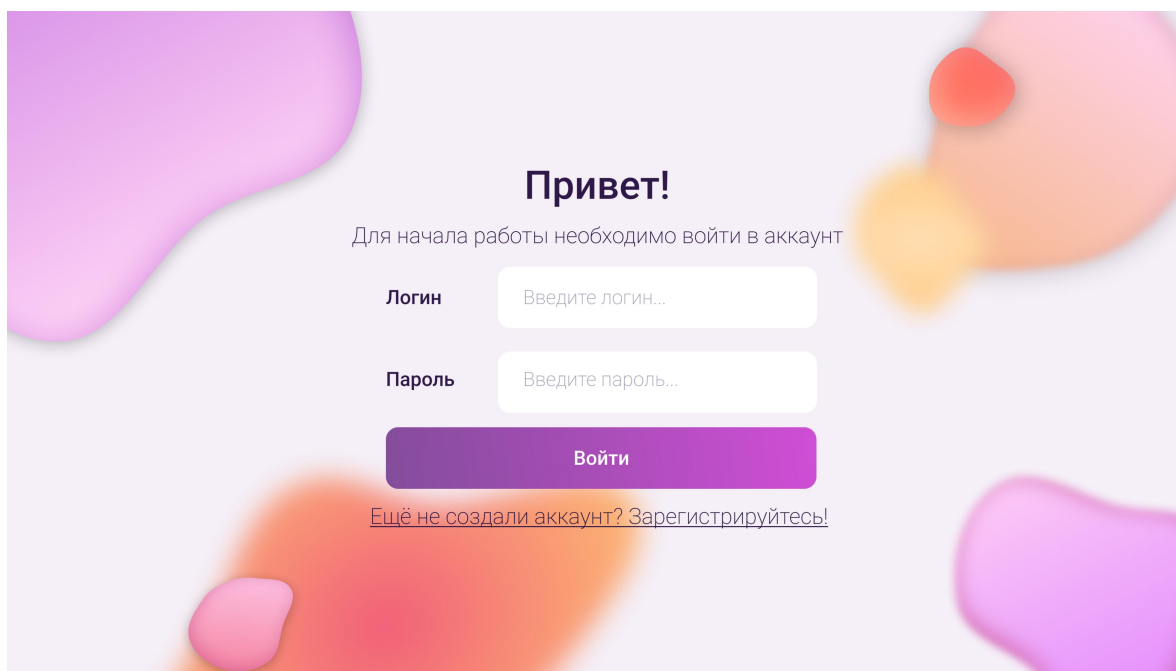


Рисунок 8 – Макет интерфейса страницы входа

В случае отсутствия у пользователя аккаунта, ему предлагается создать новый на соответствующей странице, визуализация интерфейса которой продемонстрирована на рисунке 9.

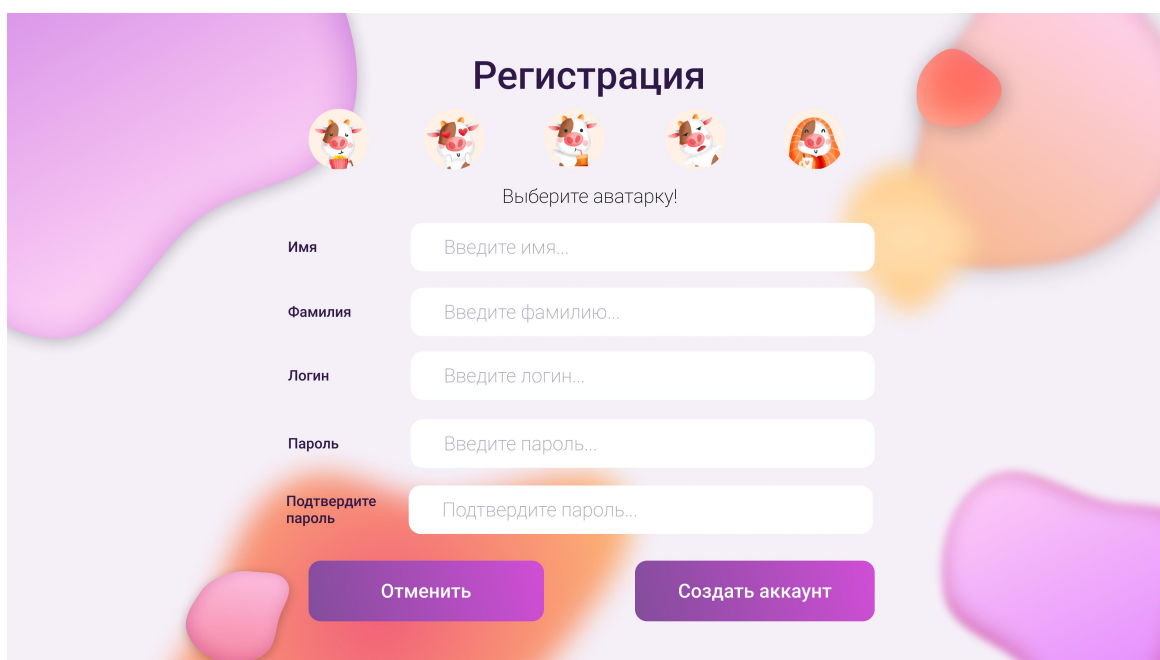


Рисунок 9 – Дизайн интерфейса страницы регистрации нового пользователя

После успешного входа в аккаунт или регистрации, пользователя отправляет на «домашнюю» страницу веб-приложения – план на текущий календарный день, где также расположены виджеты календаря и целей. Макет страницы текущего дня представлен на рисунке 10.

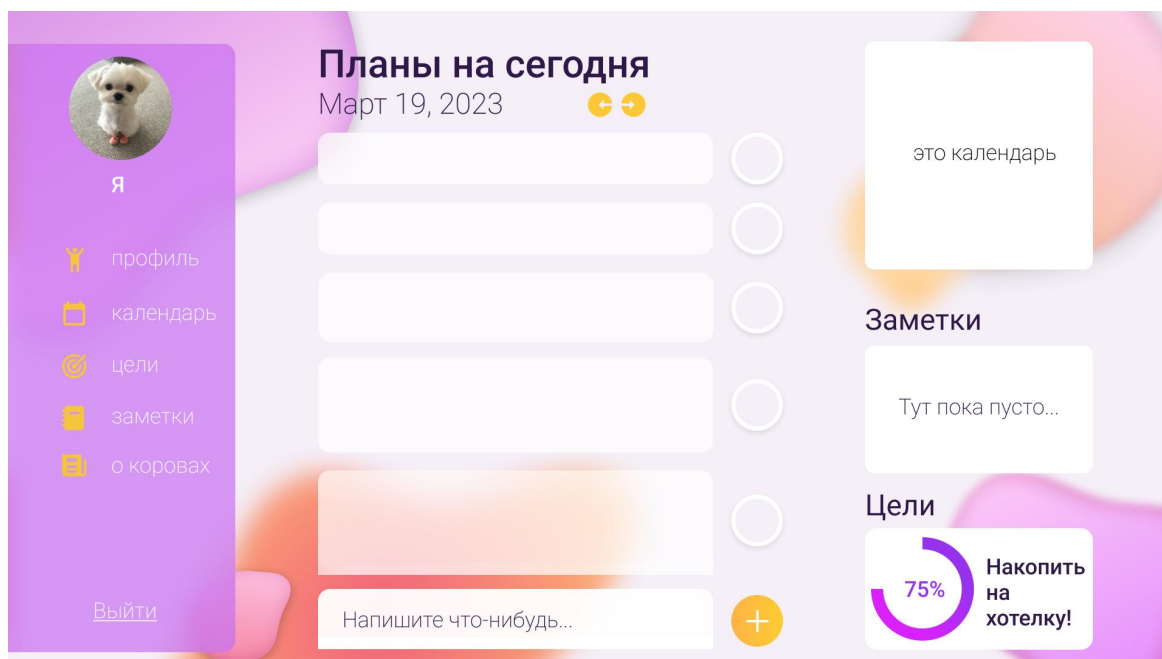


Рисунок 10 – Страница текущего дня

При переходе на вкладку «календарь» открывается страница, где изображены дела и задачи на текущую неделю, макет интерфейса

представлен на рисунке 11.

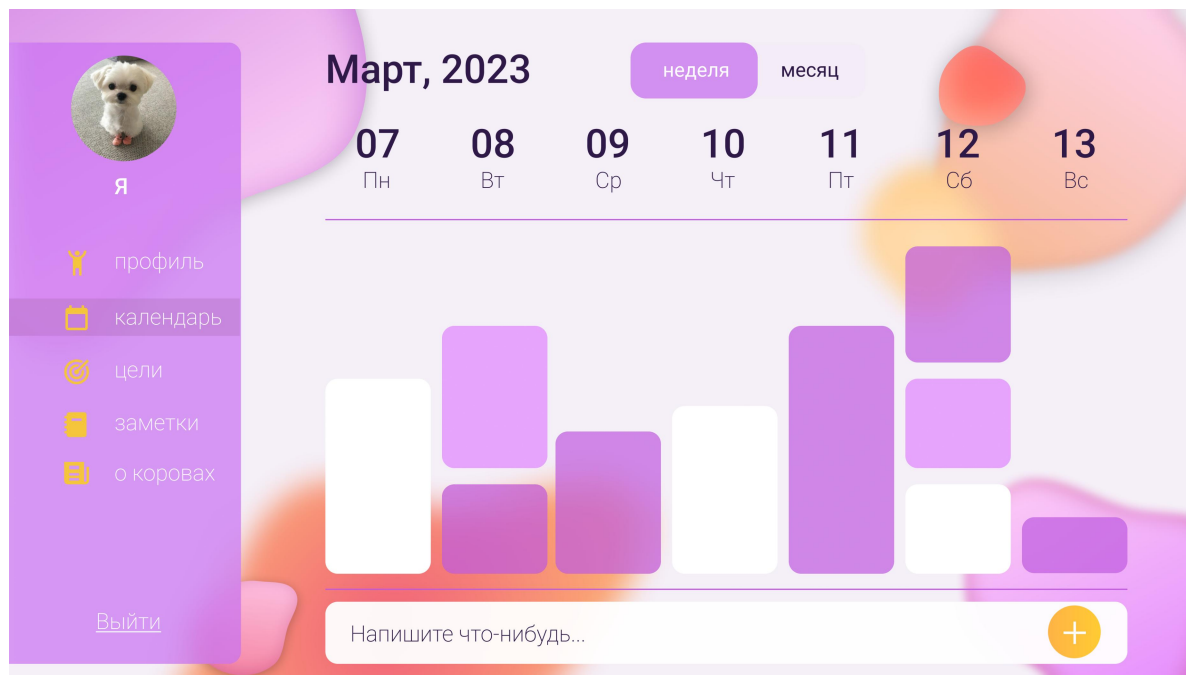


Рисунок 11 – Окно задач на текущую неделю

При нажатии соответствующей кнопки для пользователя открывается страница с календарём на месяц. Результат работы разработки макета интерфейса изображен на рисунке 12.

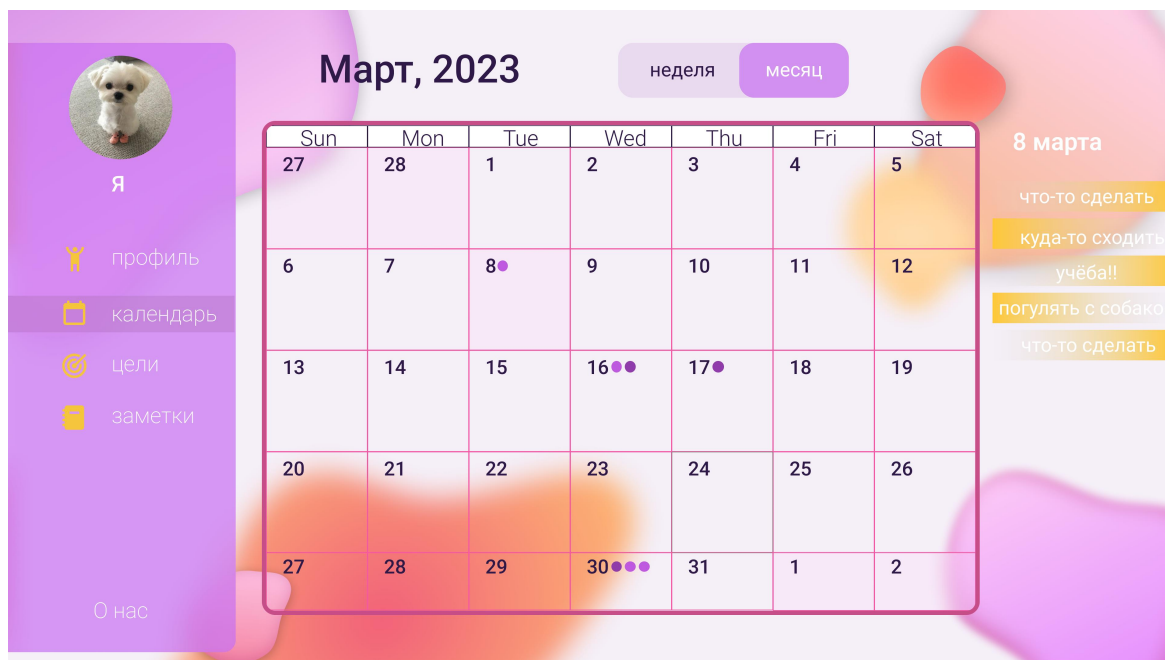


Рисунок 12 – Страница календаря

В случае нажатия в главном меню на вкладку «цели», пользователя отправляет на страницу, где отображены поставленные им цели и процент их

достижения, что продемонстрировано на рисунке 13.

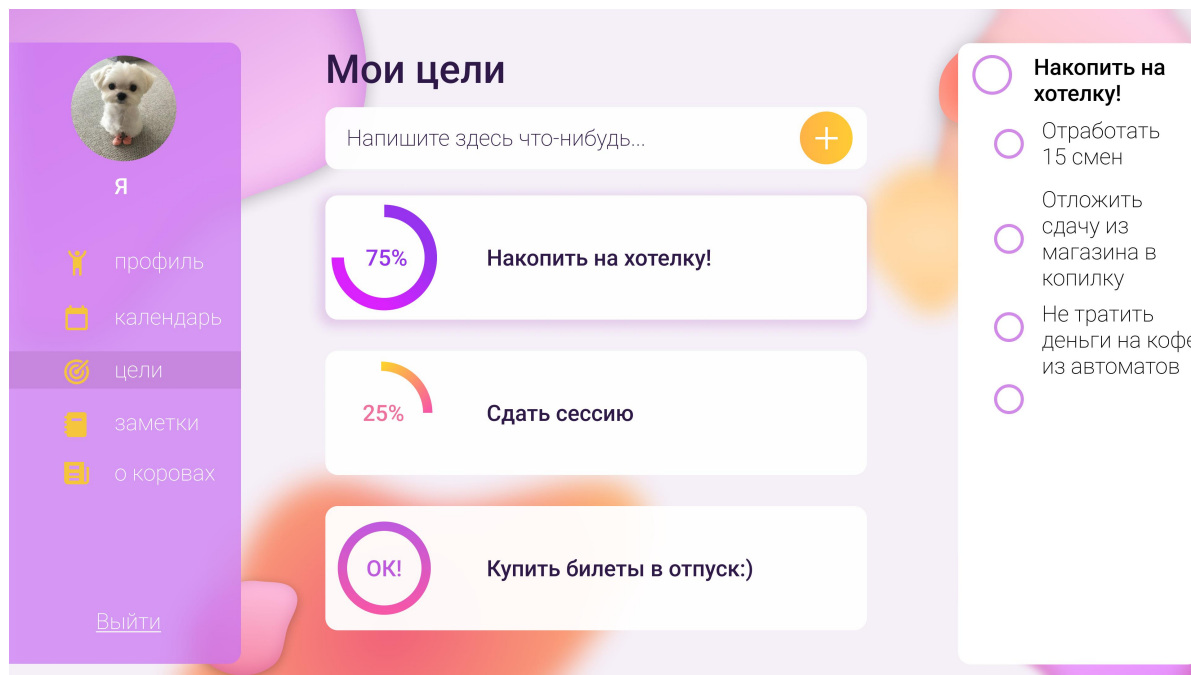


Рисунок 13 – Создание макета интерфейса страницы трекера целей

Далее, на рисунке 14 представлен макет интерфейса страницы с личными заметками пользователя.

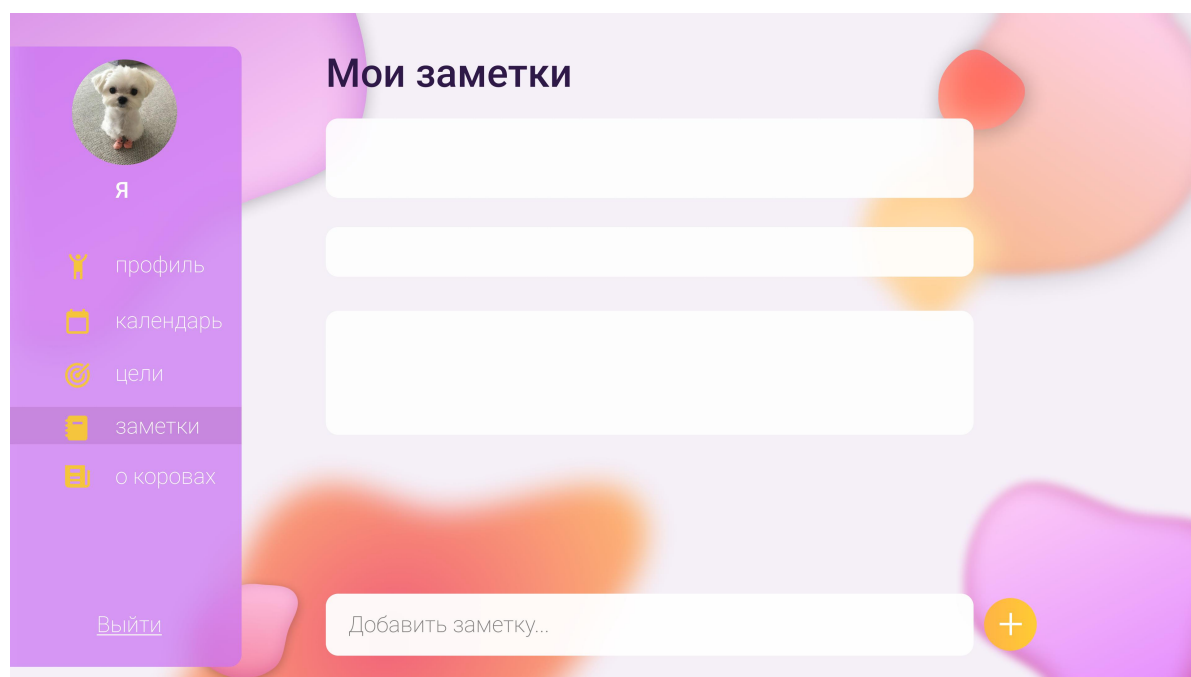


Рисунок 14 – Макет окна личных заметок

При нажатии на аватар должна открываться страница личного кабинета профиля пользователя, макет которой изображен на рисунке 15. Отсюда можно либо покинуть текущий аккаунт и перейти на страницу входа, либо

получить инструкцию по использованию приложения.

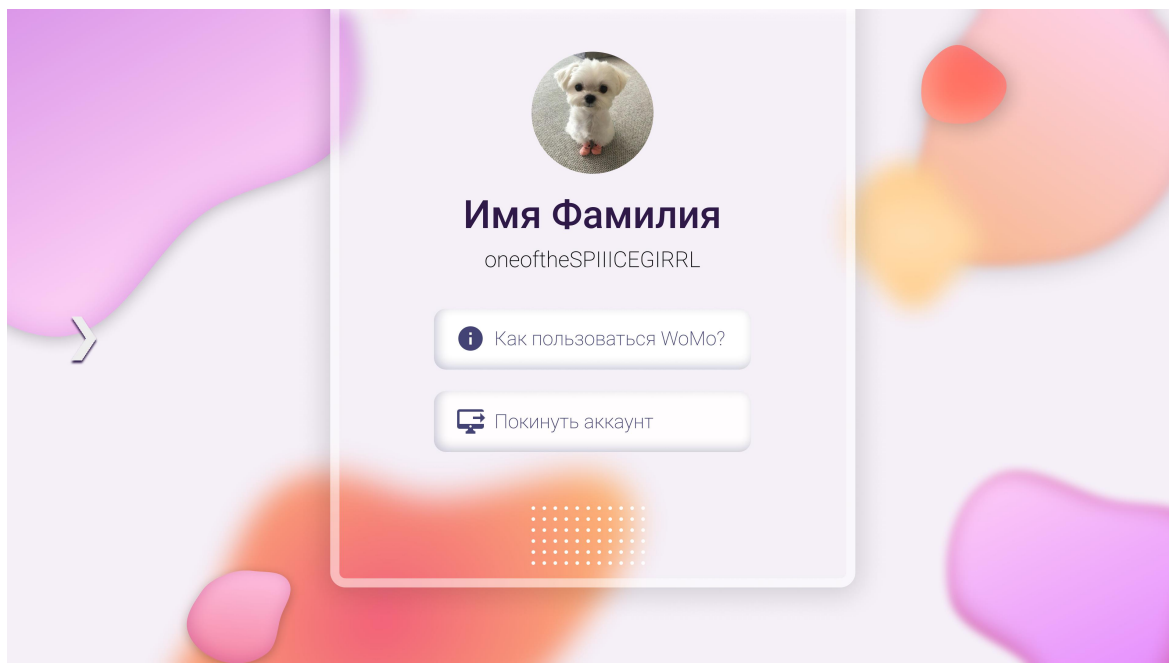


Рисунок 15 – Макет страницы личного кабинета пользователя

В ходе проектирования веб-приложения был разработан дизайн интерфейса с учетом цветовой гаммы, шрифтов, иконок и прочих элементов. Также, созданы графические элементы: иконки, кнопки, фоны для экранов и так далее. Макеты интерфейса были протестированы, результаты теста представлены на рисунке 16.

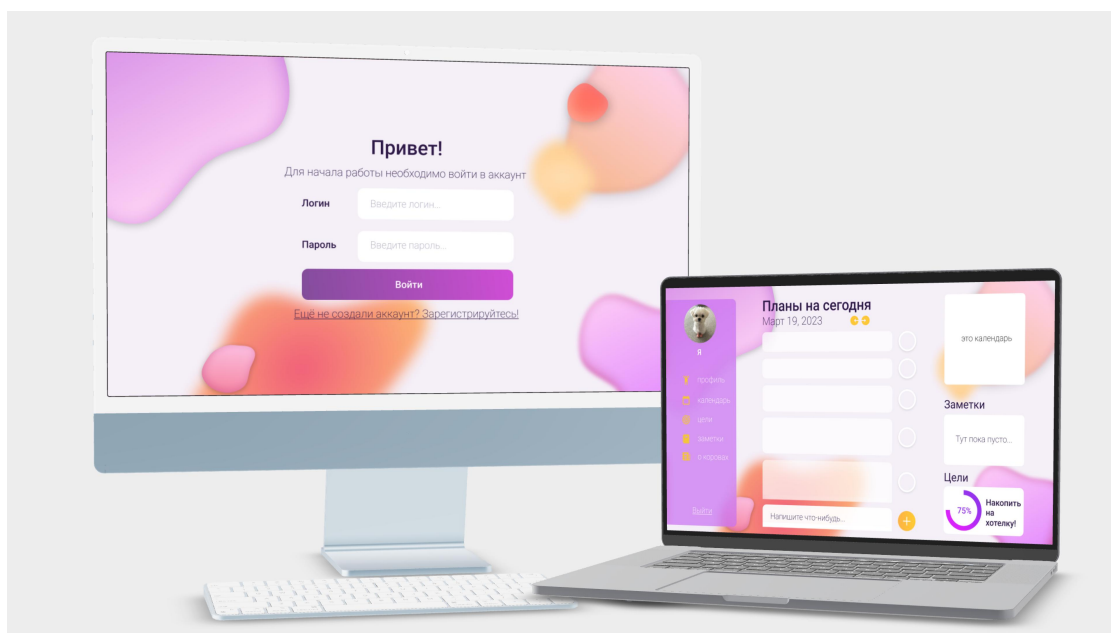


Рисунок 16 – Тестирование работоспособности макета интерфейса приложения

В ходе тестирования была собрана обратная связь, графические

элементы интегрированы с учетом принятых правок, макет интерфейса отправлен на дальнейшую реализацию.

4 Разработка клиентской части приложения

На этапе реализации разработана клиентская часть веб-приложения. На данном шаге были созданы страницы приложения с использованием HTML, CSS и JS [2].

С помощью языка HTML создана разметка сайта, утверждено положение элементов на нем. Далее с помощью CSS добавлены стили страницам сайта, оформлен внешний вид. Все функциональные возможности разработаны с помощью языка JavaScript [3].

Структура сайта:

- страница для регистрации (registration.html);
- профиль (profile.html);
- главная страница «Мой день» (index.html);
- календарь, содержащий 2 страницы: «неделя» и «месяц» (try.html, month.html);
- цели (goal_page.html);
- заметки (note_page.html).

Файлы стилей CSS имеют соответствующие названия: profile.css, newweek.css, newmonth.css, style_goal_page.css, style_note_page.css, registration.css. Дополнительные файлы для модальных окон или оставшегося интерфейса называются style.css, notes.css, side_modal.css, modal_add.css, fonts.css, menu.css.

Структура month.js:

- initCalendar() – инициализация всех элементов месяца, отображение текущего;
- prevMonth() – перелистывание на предыдущий месяц;
- nextMonth() – перелистывание на следующий месяц;
- addListener() – прослушивание нажатий пользователя на день недели, добавление дел в конкретный день, также удаление;
- getActiveDay(date) – выписывает выбранный день недели и дату;
- getWeekDayNumber(weekday) – по переданному дню недели

возвращает его номер;

— `updateEvents(date)` – аргументов является выбранная дата, записывает события в общий массив, отображает на экране список планов.

Структура файла `week.js`, содержащего функции для реализации планера на неделю.

— `initWeek()` – инициализация массива всех дней месяца, в зависимости от наличия события, актуальности месяца, также разбиение на массивы недель;

— `prevWeek()` – переключение на предыдущие недели;

— `nextWeek()` – переключение на следующие недели;

— `addListener()` – нажатие на конкретный день, возможность записи планов, удаление написанного;

— `updateEvents(date, month)` – функция обработки записанных событий в общий массив дел, также отображение дел в каждом дне, в аргументы принимает дату и месяц.

Структура `goal_create.js`:

— `displayMessages()` – вывод на экран списка подцелей в модальном окне;

— `displayHeaders()` – вывод названия целей в контейнере целей;

— `circles(which_circle, start_pos)` – по переданному положению высчитывает процент выполнения цели.

Структура `memo_for_bd.js`:

— `setText(param1, param2, param3)` – заполняет определенный блок структуры данных;

— `viewNote(param)` – выводит на экран созданную заметку.

Структура `memo.js`:

— `createNote(title, text)` – функция для создания заметки;

— `displayMessages()` – отображение заметок на странице.

Структура `my_day.js`:

— `displayMessages()` – отображение дел на день в контейнере планов;

- `main_circle()` – отображение последней цели на главной странице;
- `main_notes()` – отображение последней заметки на главной странице.

5 Разработка серверной части приложения

В процессе реализации веб-приложения также разработана серверная часть. На этом этапе создается серверное приложение на языке Python с использованием фреймворка Django [4]. В приложении определяются модели базы данных, создаются контроллеры и представления.

Проект состоит из пяти приложений, организующих слаженную работу итогового веб-приложения. Django создает базовую структуру для каждой новой части программы. Структура состоит из:

- файла `admin.py` - здесь зарегистрированы модели, чтобы включить их в административную часть сайта Django;
- `migrations/` - этот каталог содержит миграции базы данных приложения. Миграция позволяет отслеживать изменения модели и соответствующим образом синхронизировать базу данных;
- `models.py` - модели данных приложения. Для всех приложений требуется `models.py` файл;
- `views.py` - логика приложения здесь. Каждое представление(`view`) получает HTTP-запрос, обрабатывает его и возвращает ответ.

Помимо базовых файлов дополнительно были добавлены директории `templates` для хранения HTML файлов и `static` для хранения изображений, шрифтов, файлов CSS и JavaScript. Созданы файлы `urls` для удобного хранения и использования URL-адресов каждого приложения. Создан базовый HTML шаблон для дальнейшего расширения в разных частях программы.

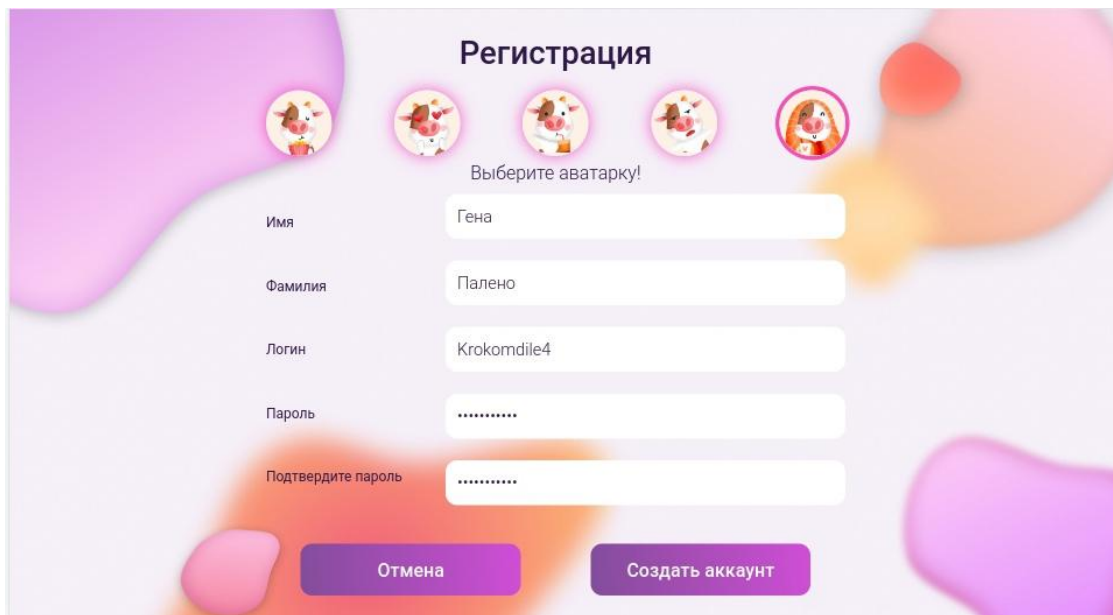
Отправка данных клиенту происходит путем получения из соответствующей базы данных информации, конвертации ее в необходимую структуру данных и их запись в формате JSON. Впоследствии этот файл отправляется с помощью инструментов, предоставленных фреймворком [5].

Исключением стало приложение регистрации. Для получения информации были использованы формы — наборы полей, предоставляющих данные объектов.

6 Тестирование

На этом этапе проводятся тесты приложения, исправляются ошибки и устраняются неполадки. После тестирования и отладки приложение готово к релизу. Оно может быть развернуто на сервере и использоваться пользователями.

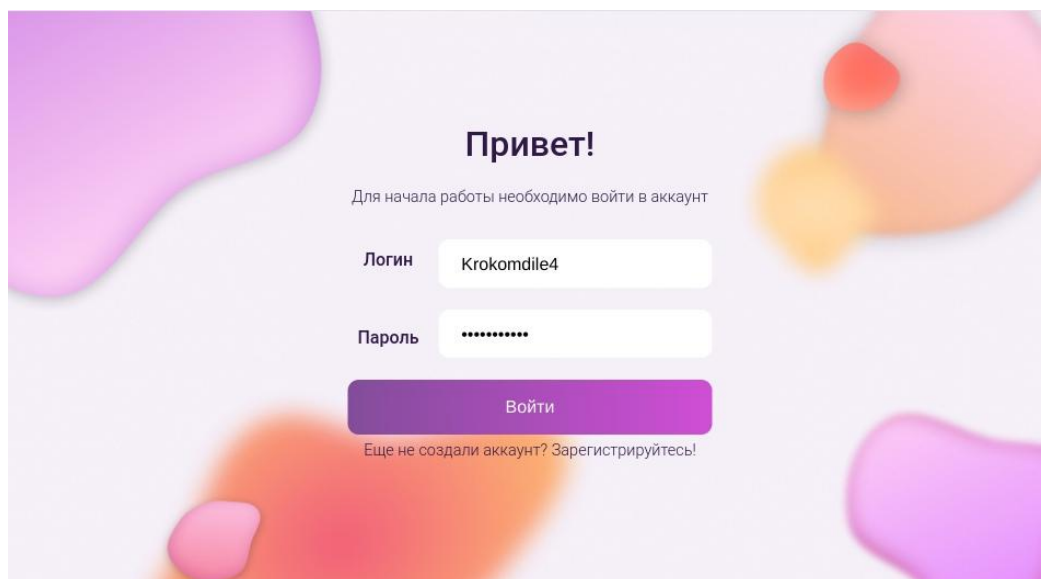
На первом этапе пользователю представляется возможность зарегистрировать свой аккаунт, как представлено на рисунке 17.



The screenshot shows a registration form with a light purple background and abstract pink and orange shapes. At the top, the title 'Регистрация' is centered. Below it, there are five circular avatars of a cartoon character, with the text 'Выберите аватарку!' (Choose an avatar!) above them. The form contains five input fields: 'Имя' (Name) with the value 'Гена', 'Фамилия' (Surname) with 'Палено', 'Логин' (Login) with 'Krokomdile4', 'Пароль' (Password) with masked characters, and 'Подтвердите пароль' (Confirm password) with masked characters. At the bottom, there are two buttons: 'Отмена' (Cancel) in a light purple box and 'Создать аккаунт' (Create account) in a darker purple box.

Рисунок 17 – Регистрация пользователя

Затем, как показано на рисунке 18, пользователь сможет войти в аккаунт под своим логином.



The screenshot shows a login form with the same light purple background and abstract shapes. The title 'Привет!' (Hello!) is centered at the top. Below it, the text 'Для начала работы необходимо войти в аккаунт' (To start working, you need to log in to your account) is displayed. The form has two input fields: 'Логин' (Login) with the value 'Krokomdile4' and 'Пароль' (Password) with masked characters. Below these fields is a single purple button labeled 'Войти' (Log in). At the bottom, there is a link that says 'Еще не создали аккаунт? Зарегистрируйтесь!' (Don't have an account yet? Register!).

Рисунок 18 – Вход в аккаунт

При входе открывается главная страница, на которой человек может ввести планы на день, как на рисунке 19.

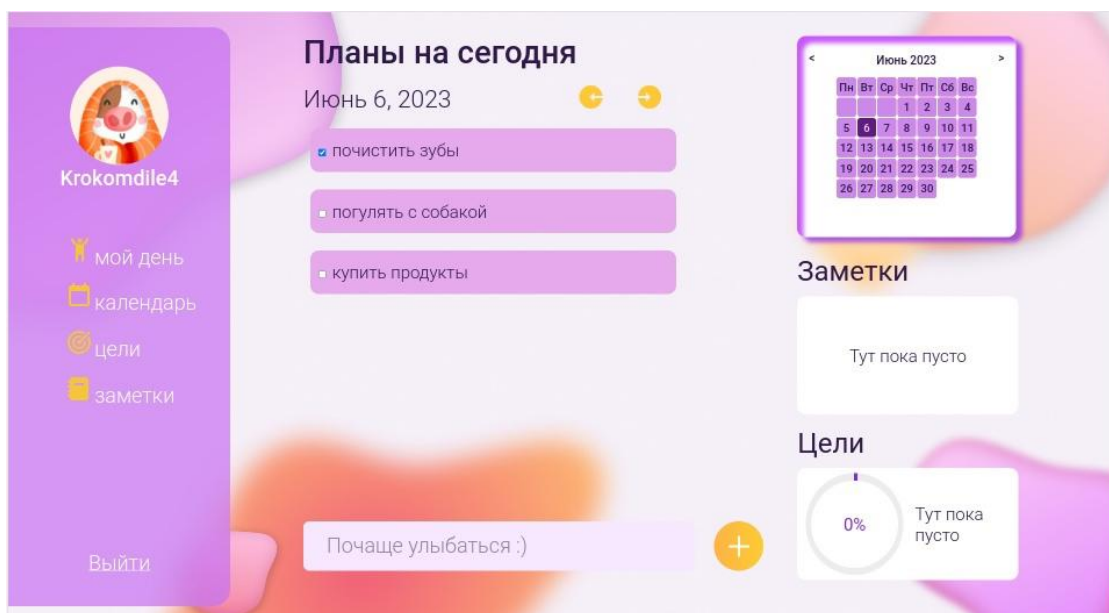


Рисунок 19 – Главная страница

Далее, в вкладке «календарь» появятся введенные ранее дела на день, как продемонстрировано на рисунке 20.

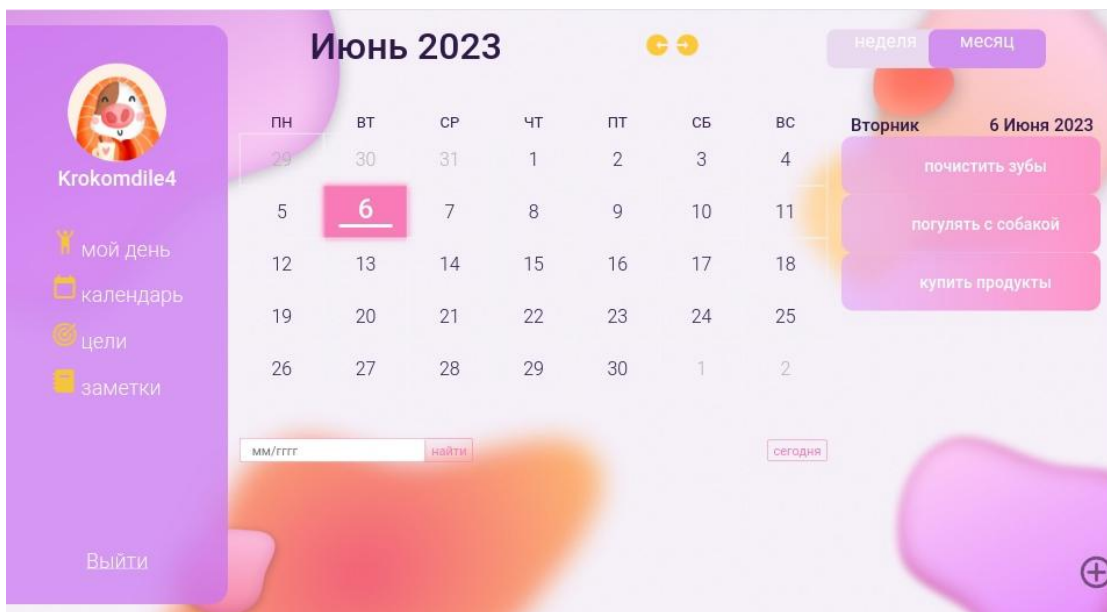


Рисунок 20 – Планер на месяц

Также есть возможность перейти на вкладку «неделя», чтобы детальнее распланировать любую неделю, как показано на рисунке 21.

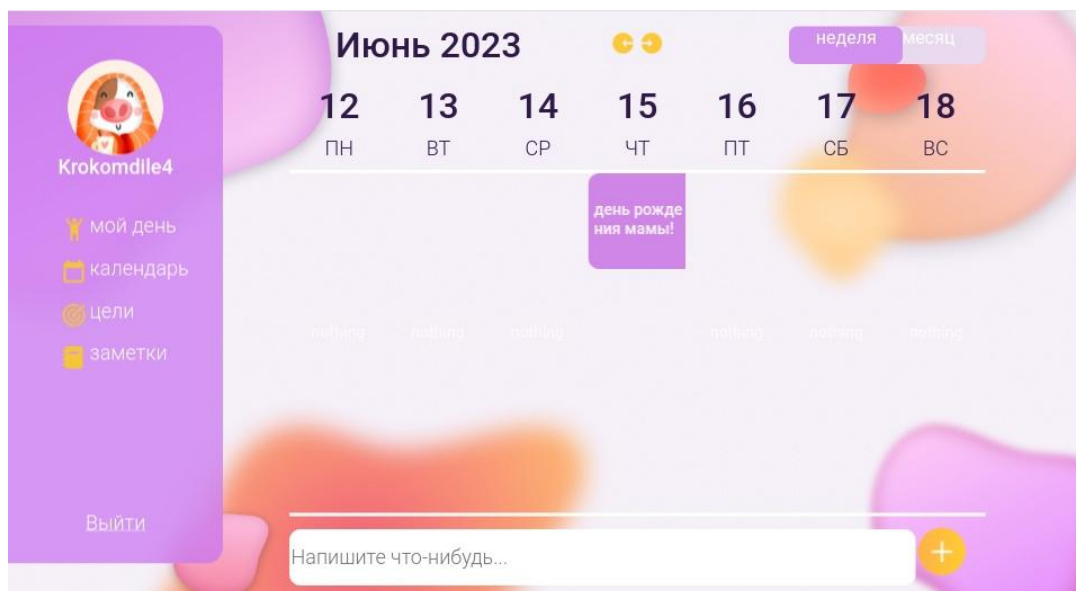


Рисунок 21 – Планер на неделю

На рисунке 22 изображено добавление целей.

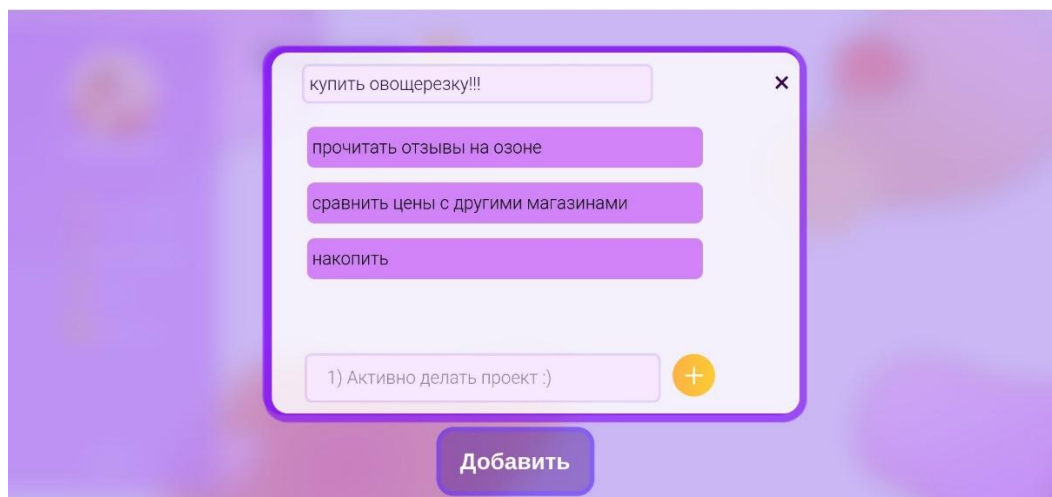


Рисунок 22 – Добавление целей

На рисунке 23 представлено, как выполнение подцелей влияет на состояние трекера.

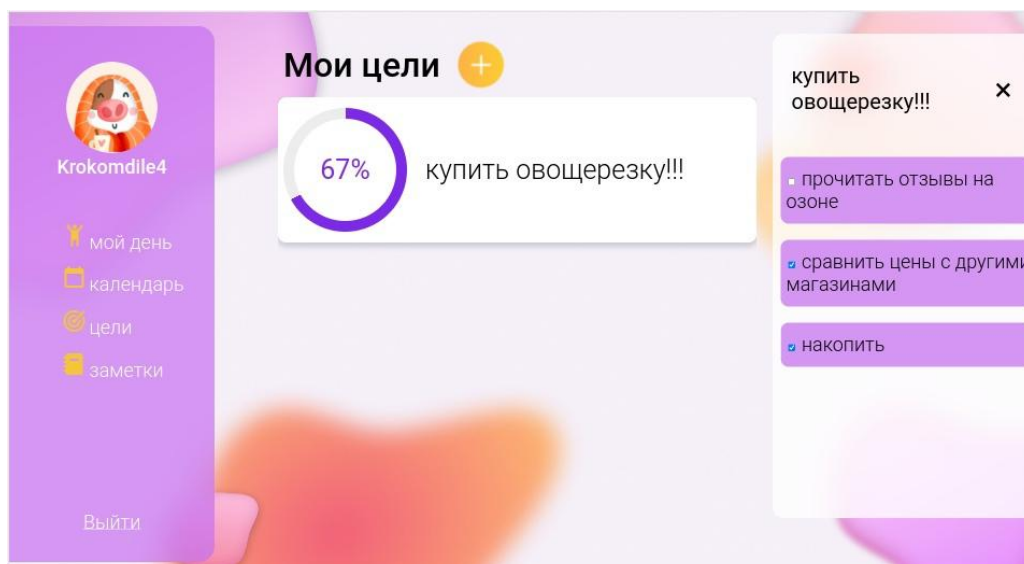


Рисунок 23 – Страница целей

Заметки располагаются списком на экране, как представлено на рисунке 24.

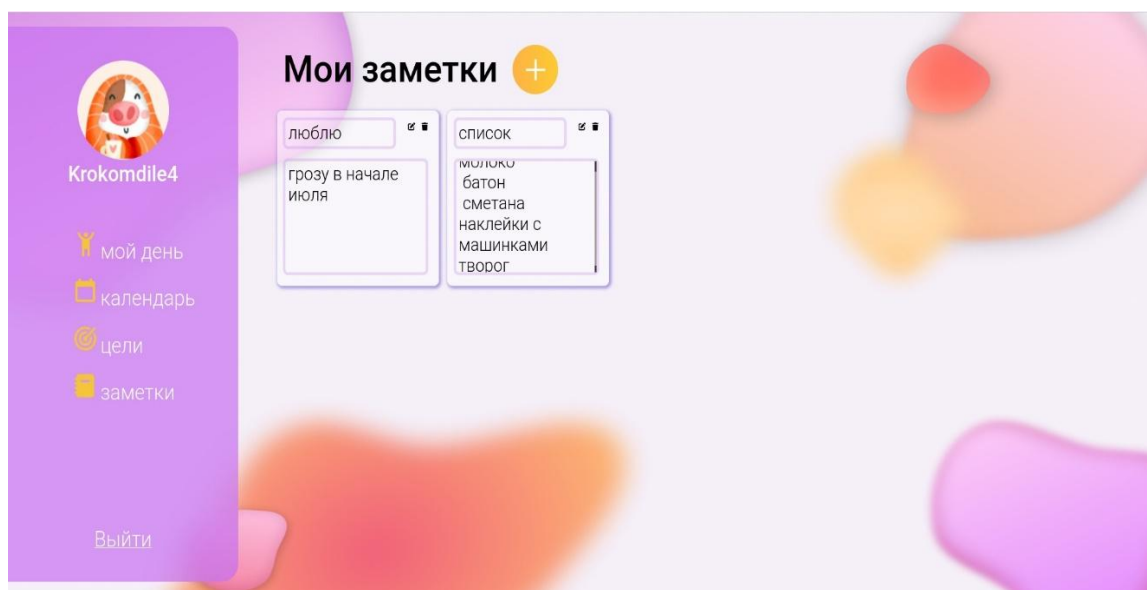


Рисунок 24 – Заметки

Также у пользователя есть возможность посетить свой профиль, в котором он может обратиться к инструкции по сайту либо выйти из аккаунта, как представлено на рисунке 25.

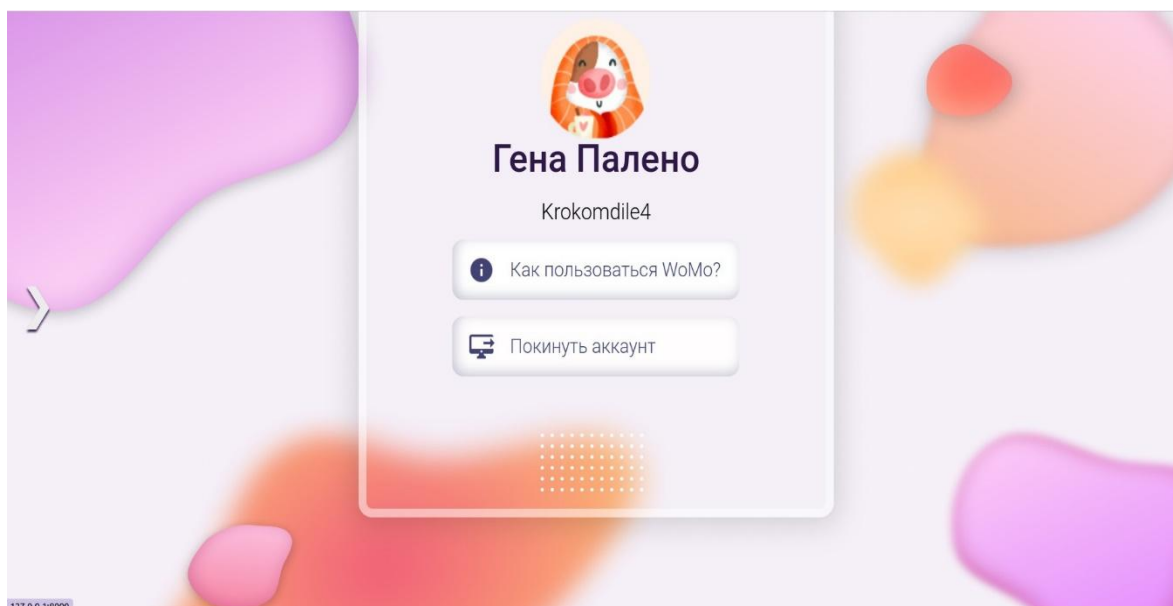


Рисунок 25 – Профиль

ЗАКЛЮЧЕНИЕ

В данной проектной работе было разработано кросс браузерное веб-приложение WoMo, способное управлять задачами и планировать расписание пользователя.

Исходя из поставленных задач:

- Изучены особенности различных браузеров и их поддержки технологий;
- Разработан дизайн и интерфейс приложения;
 - база данных для хранения информации о задачах;
- Разработан функционал для добавления, редактирования и удаления задач;
- Разработаны дополнительные виджеты.

Данная версия приложения может служить основой для создания более сложных версий.