

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г.ЧЕРНЫШЕВСКОГО»**

Кафедра Математического и компьютерного моделирования

Проектирование и реализация финансового приложения

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студента 5 курса 561 группы

направление 09.03.03 — Прикладная информатика

механико-математического факультета

Деревянкина Данила Анатольевича

Научный руководитель
доцент, к.ф.-м.н., доцент

О.С. Кузнецова

Зав. кафедрой
зав. каф., д.ф.-м.н., доцент

Ю.А. Блинков

Саратов 2025

Введение. Учёт доходов и расходов, планирование затрат, постановка целей — всё это помогает привести финансовое состояние в порядок и выделить средства для инвестирования. Способов контроля денежных ресурсов достаточно: запись в тетрадь, заполнение таблиц в Excel, использование профильных программ для ПК и смартфонов. Последнее — лучший вариант того, как быстро и удобно вести учёт домашних расходов. Для этого было разработано приложение Finance App.

Основной целью данной работы служит разработка приложения, с помощью которого можно облегчить процесс расчета и ведения бюджета.

Приложение Finance App позволяет выяснить, на что, на самом деле, пользователь тратите деньги. Зачастую, учет финансов занятие не из приятных и требует конкретных навыков и умений. Оно позволяет легко вести учет расходов и доходов, составлять бюджет, помогает быстрее рассчитаться с долгами, делать сбережения денег и достигать финансовых целей.

Темой данной работы является проектирование и реализация финансового приложения, так же учет средств бюджета. Данная тема близка каждому. У любого пользователя есть свои доходы и расходы. Естественно возникает необходимость учета финансовых средств.

Для ознакомление с приложением Finance App, вы можете перейти по этой ссылки <https://keplcas.github.io/FinanceApp/>.

Основное содержание работы. Для разработки приложения были выбраны следующие стек-технологии: TypeScript, React, Mobx StateTree, Styled Components, React Hook Form, Date FNS, chart.js, Evergreen UI, axios, NODE.

Авторизация. В соответствии с рисунком 1, представлено окно авторизации, которое отображается при входе в приложение.

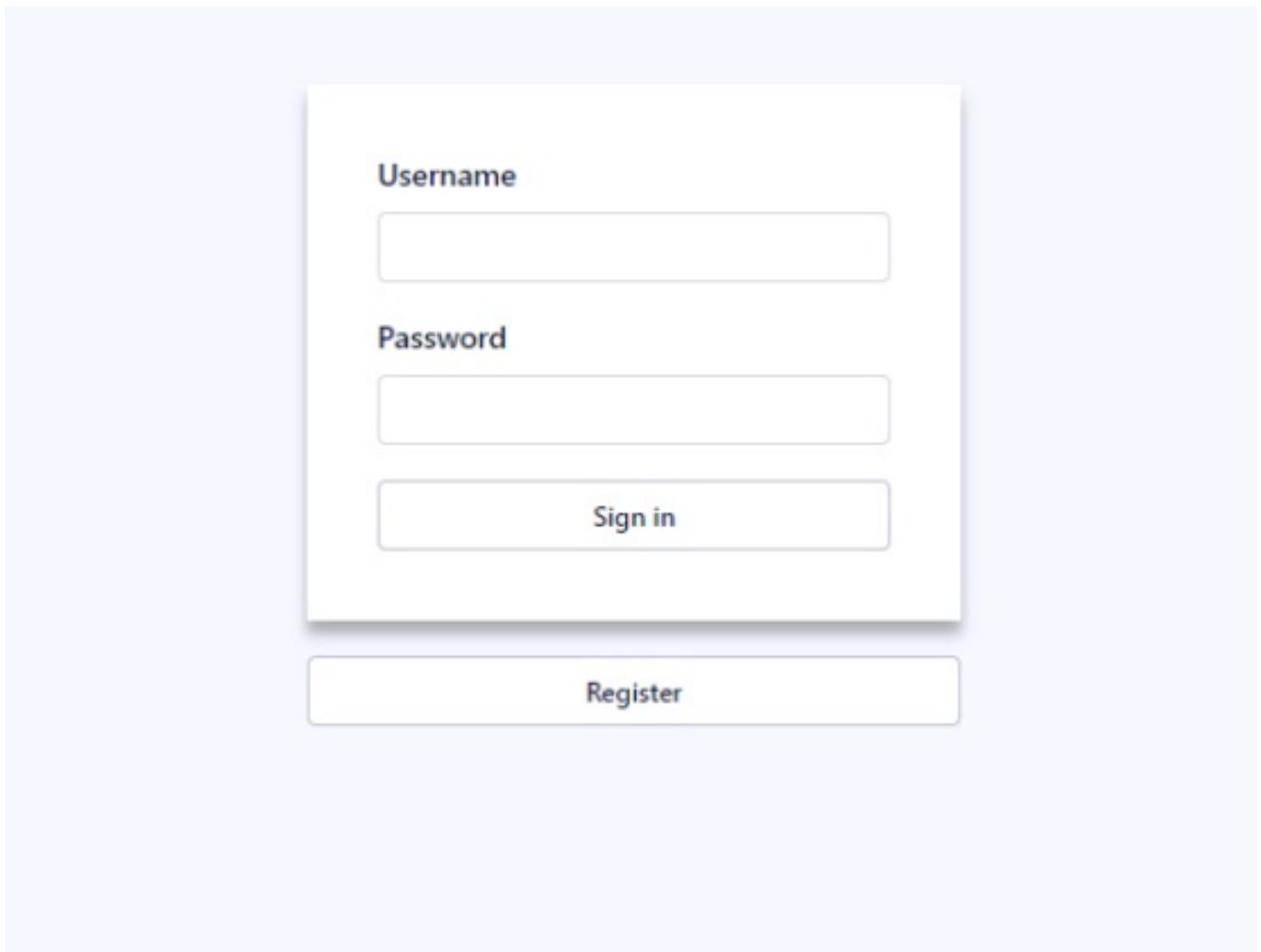


Рисунок 1 — Окно авторизации

Авторизация написана на Express, Mongoose и MongoDB.

После регистрации появится приветственное окно, представленное в соответствии с рисунком 2

Главная страница. В соответствии с рисунком 3, представлена главная страница разрабатываемого приложения.

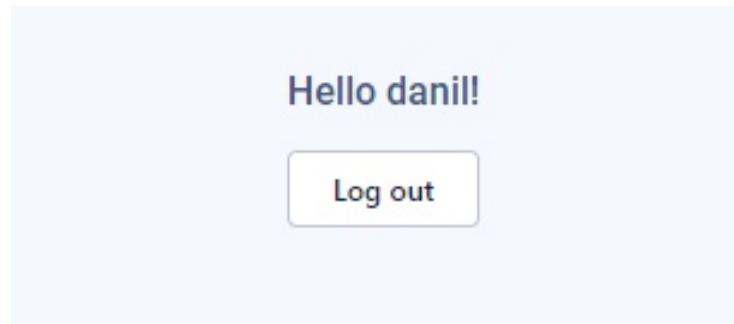


Рисунок 2 — Приветственное окно

ID	Date	Category	Type	Quantity
625bdec...	17/04/2022	Salary	Income	5000
625befc...	12/04/2022	Transport	Expenses	500
625befc...	17/04/2022	Transport	Expenses	500
625befc...	17/04/2022	Credits	Expenses	200
625befc...	17/04/2022	Gifts	Expenses	100

Рисунок 3 — Главная страница

На главной странице отображается таблица транзакций, добавление транзакции, фильтрация по категориям. Начнем с добавление транзакции, чтобы добавить новую транзакцию необходимо нажать на кнопку с плюсом и открыть модальное окно. В соответствии с рисунком 4 представлено модальное окно добавления транзакции, в котором пользователь выбирает дату транзакции, категорию, добавляет описание (не обязательно), выбирает доход или расход и вводит сумму.

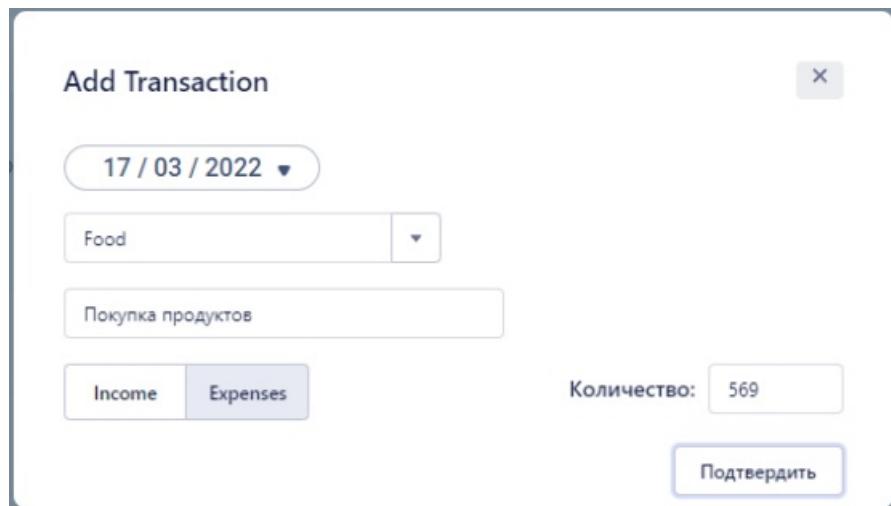


Рисунок 4 — Главная страница

После подтверждения, модальное окно закрывается, и транзакция отображается в таблице, представленной в соответствии с рисунком 5.

Transactions list			<input type="checkbox"/> Income	<input type="checkbox"/> Expenses	From	01 / 03 / 2022 *	To	30 / 04 / 2022 *
<input type="checkbox"/>	ID	Date	Category	Type	Quantity	<input type="button" value="Delete"/>	<input type="button" value="Edit"/>	
<input type="checkbox"/>	625bdec...	17/04/2022	Salary	Income	5000	<input type="button" value="Delete"/>	<input type="button" value="Edit"/>	
<input type="checkbox"/>	625befce...	12/04/2022	Transport	Expenses	500	<input type="button" value="Delete"/>	<input type="button" value="Edit"/>	
<input type="checkbox"/>	625befce...	17/04/2022	Transport	Expenses	500	<input type="button" value="Delete"/>	<input type="button" value="Edit"/>	
<input type="checkbox"/>	625befef...	17/04/2022	Credits	Expenses	200	<input type="button" value="Delete"/>	<input type="button" value="Edit"/>	
<input type="checkbox"/>	625bef6...	17/04/2022	Gifts	Expenses	100	<input type="button" value="Delete"/>	<input type="button" value="Edit"/>	
<input type="checkbox"/>	625c20c4...	17/03/2022	Food	Expenses	569	<input type="button" value="Delete"/>	<input type="button" value="Edit"/>	

Рисунок 5 — Таблица с транзакциями

Пользователь может нажать на транзакцию и посмотреть ее данные, а также редактировать или удалить ее. В соответствии с рисунком 6 представлено окно настроек транзакции, через которое можно произвести перечисленные ранее манипуляции.

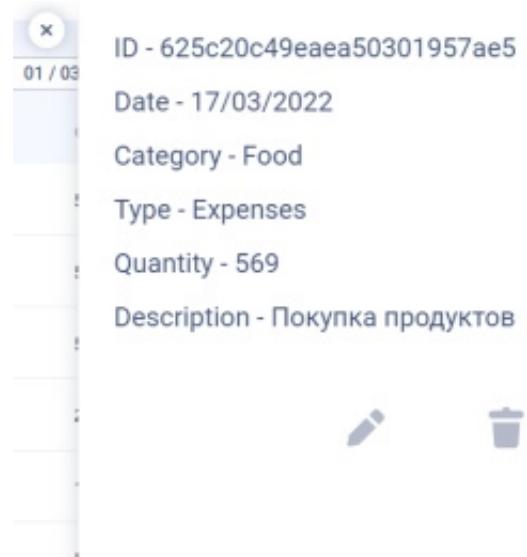


Рисунок 6 — Редактирование транзакции

Так же пользователь может фильтровать по категориям, как это представлено в соответствии с рисунком 7, что упрощает просмотр таблицы.

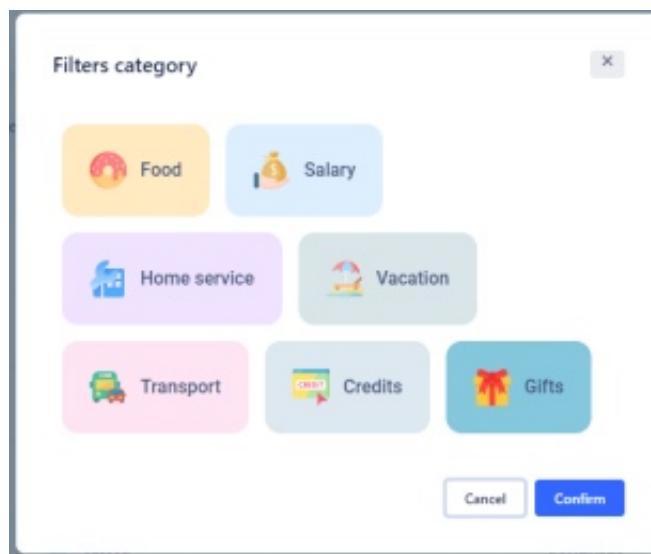


Рисунок 7 — Категории

Страница информации. На странице информации пользователь может посмотреть доступную информацию в гистограмме и круговой диаграмме.

В гистограмме можно выбрать тип дохода или расхода и посмотреть на данные. В соответствии с рисунком 8 представлена гистограмма.

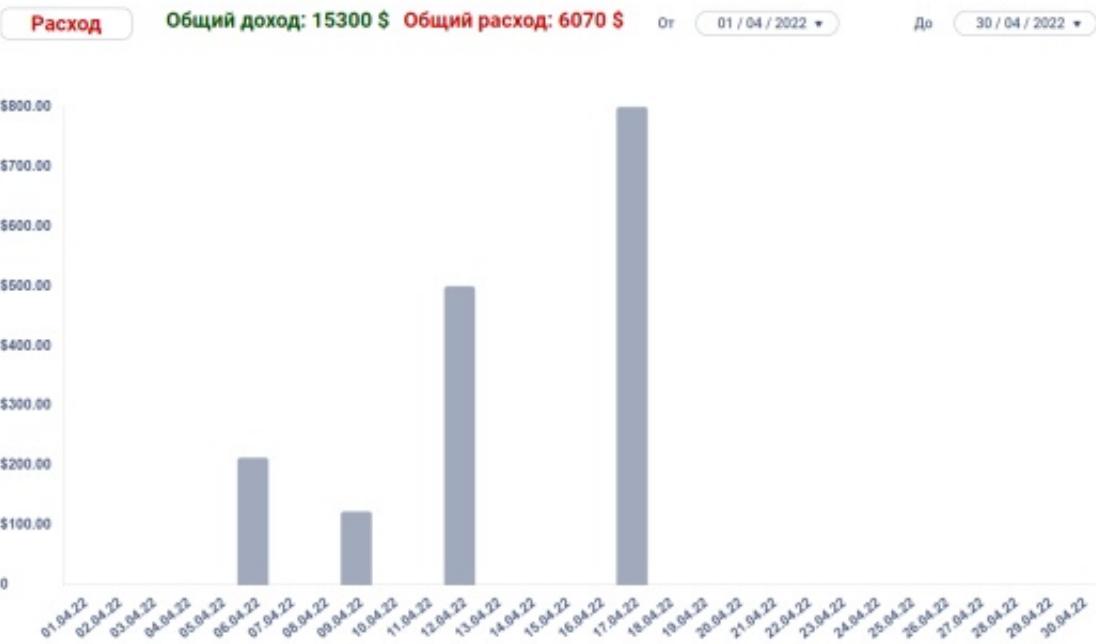


Рисунок 8 — Гистограмма

Также приложение позволяет выбрать даты, попадающие в указанный период, для последующего отображения, представленного в соответствии с рисунком 9. Эта возможность была реализована при помощи библиотеки DATE-FNS, позволяющей работать с датами.

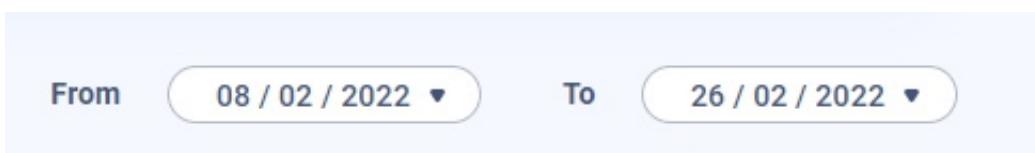


Рисунок 9 — Выбор периода отображения

Помимо гистограммы, данные могут быть отображены в виде круговой диаграммы, где данные о расходах разделены на сегменты, а дуга каждого сегмента показывает пропорциональное значение каждого фрагмента данных. Они отлично показывают относительные пропорции между данными. В соответствии с рисунком 10 представлена круговая диаграмма, в которой отображаются расходы пользователя.

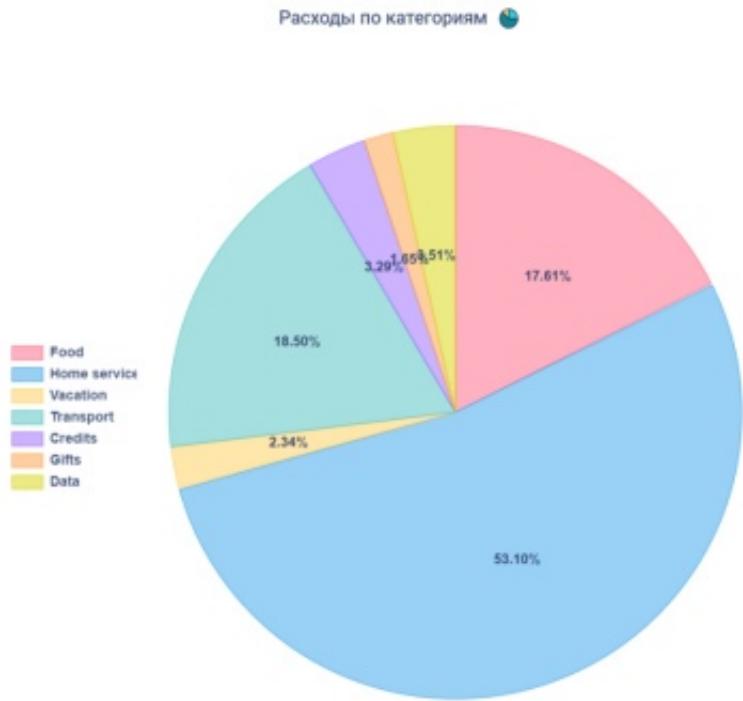


Рисунок 10 — Круговая диаграмма

Дополнительные преимущества. Данное приложение поддерживает помимо русского языка еще и английский, что способствует увеличению пользовательской аудитории. Для перехода на другой язык достаточно кликнуть по кнопки, представленной в соответствии с рисунком 11.



Рисунок 11 — Переключатель языка

Для возможности выбора языка приложения была использована библиотека **FormatJS**. FormatJS — это модульная коллекция библиотек JavaScript для интернационализации, ориентированная на форматирование чисел, дат и строк для отображения людям. Он включает в себя набор основных библиотек, основанных на встроенных модулях JavaScript Intl и отраслевых стандартах i18n, а также набор интеграций для общих библиотек шаблонов и компонентов. В соответствии с рисунком 12 представлена архитектура i18n.

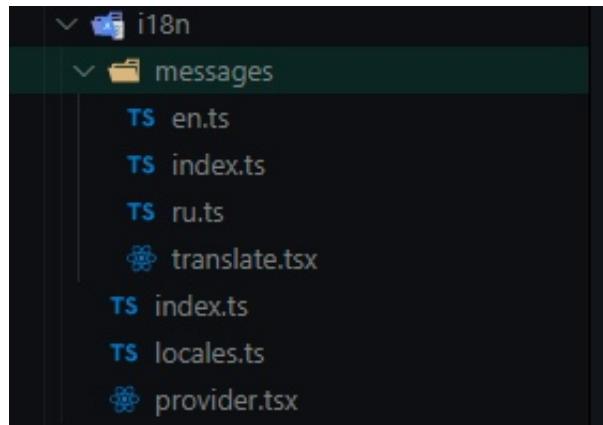


Рисунок 12 — Архитектура i18n

Index.ts является главным файлом, куда импортируется locales.ts и provider.tsx. В locales.ts создается и экспортируется переменная LOCALES.

В provider.tsx оберачивает в IntlProvider, React Intl использует шаблон провайдера, чтобы ограничить контекст i18n деревом компонентов. Это позволяет предоставлять такую конфигурацию, как текущая локаль и набор переведенных строк/сообщений, в корне дерева компонентов и делать их доступными для <Formatted*> компонентов. Далее файл en.ts и ru.ts содержит в себе строки/сообщения используемые в приложение.

Помимо этого, пользователь может воспользоваться «темной темой», которая экономит батарею аккумуляторов с OLED - дисплеем, комфорт для глаз и фокусировка на задачах. Для перехода на «темную тему» достаточно кликнуть по переключателю, представленной в соответствии с рисунком 13.

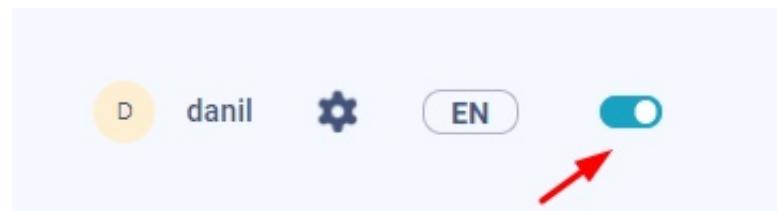


Рисунок 13 — Переключатель темы

Смена светлой темы на темную, происходит следующим образом. Для начала создаем объект и внутри объекта создаём два ключа: light, dark. И добавляем им в значения объект цветов со светлой и темной темы, как это представлено в соответствии с рисунком 14

```
const themeColours = {
  light: {
    color: "#435379",
    backgroundColor: "#F5F8FF",
    colorLight: "rgba(67, 83, 121, 0.4)",
    backgroundColorBlock: "#ffffff",
    colorInc: "#004d00",
    colorExp: "#cc0000",
    nightBlue: "#101840",
    bgCalendar: "#fffffa",
    colorWhite: "#ffffff",
    pagination: "#fff",
  },
  dark: {
    color: "#777777",
    backgroundColor: "#18181B",
    colorLight: "rgba(89, 89, 89, 0.7)",
    backgroundColorBlock: "#272B35",
    colorInc: "#079107",
    colorExp: "#db2020",
    nightBlue: "#C0C7E7",
    bgCalendar: "#6883c2",
    colorWhite: "#000",
    pagination: "#C0C7E7",
  },
};
```

Рисунок 14 – Объект цветовой настройки

Далее создаем стейт, где значение по умолчанию будет светлая тема.

Далее оборачиваем все наше приложение в `ThemeProvider`. Чтобы переключиться на темную тему, создаем переключатель.

Заключение. Сведения, содержащиеся в работе, обладают практической значимостью для всех, тем самым облегчает процесс расчета и ведения бюджета, так как некоторые люди сталкиваются с проблемой правильного распределения денежных ресурсов.

Использование данного приложения в реальных условиях позволяет повысить уровень финансовой грамотности, так как данное приложение позволяет получить целостную картину доходов и расходов.