

Планирование и учет личного бюджета с использованием API искусственного интеллекта на python

Научный руководитель – **Андреянов Никита Андреевич**

Новик Ольга Сергеевна

Студент (магистр)

Российская академия народного хозяйства и государственной службы при Президенте Российской Федерации, Институт прикладных экономических исследований, Москва, Россия

E-mail: ranet9711@mail.ru

Аннотация/введение. Личный бюджет часто ведется нерегулярно: данные о расходах и доходах фиксируются с задержкой, категории подбираются «на глаз», а планирование превращается в набор общих советов. На фоне роста безналичных платежей и подписочной экономики пользователю важно получать оперативную картину по фактическим тратам и прогнозу до конца месяца. Использование API искусственного интеллекта позволяет упростить ввод данных на естественном языке и сделать рекомендации более адресными, но при этом требуется сохранять точность учета и проверяемость расчетов.

Цель и задачи. Цель работы — предложить концепцию приложения на Python для планирования и учета личного бюджета, в котором ИИ через API отвечает за понимание пользовательских сообщений, а расчеты выполняются детерминированным «движком бюджета». Для достижения цели решались задачи: (1) определить поток обработки событий (расход/доход/цель/лимит), (2) описать модель контекста, которая поддерживает актуальность советов во времени, (3) предложить механизм валидации и уточнений, снижающий риск ошибок при автоматическом разборе текста.

Методы. В работе использованы методы проектирования информационных систем и прототипирования. Предложена событийная модель учета (ledger): каждое сообщение пользователя превращается в структурированное событие и сохраняется в базе данных (SQLite/PostgreSQL). Для извлечения данных из текста применяются структурированные ответы по JSON-схеме (сумма, валюта, категория, дата, признак необходимости уточнения). Для взаимодействия клиента и сервера рассматривается реализация API на FastAPI. Адаптивность планирования достигается через регулярное обновление «снимков бюджета» и правил пользователя (стратегия, ограничения, предпочтения по стилю подсказок), а также через учет обратной связи о полезности рекомендаций.

Результаты. Разработана логика работы приложения, разделяющая «смысл» и «цифры»: ИИ выступает диспетчером, который распознает намерение пользователя и формирует структуру события, а Python-движок является источником истины для сумм, лимитов и прогнозов. Поток обработки включает этапы: прием сообщения; определение типа события; извлечение структуры; валидация (проверка валюты, даты, подозрительных сумм, повторов); запись в учет; пересчет показателей (факт, остатки по категориям, прогноз до конца периода, ближайшие обязательные платежи); формирование ответа.

Предложены варианты контекстных сообщений, которые подмешиваются к запросам ИИ: (а) мини-контекст парсера (валюта по умолчанию, допустимые категории, правило «не уверена — спрашивай»), (б) снимок бюджета за текущий период (факт, лимиты, обязательные платежи, цели), (в) стратегия пользователя (приоритеты и запреты), (г) список неопределенностей (что нужно уточнить), (д) метрики полезности (какие советы были приняты и дали эффект). Контекст обновляется по времени: числовые данные пересчитываются при каждой операции, неопределенности имеют срок актуальности, а стратегия

и предпочтения версионироваться при изменениях.

На базе концепции показан MVP-подход: единая точка ввода (/ingest) принимает текст, ИИ возвращает структурированное событие по схеме, затем сервер сохраняет его и возвращает пользователю короткое подтверждение и обновленный снимок бюджета. Такой подход сохраняет точность учета и снижает нагрузку на пользователя.

Выводы. Предложенная концепция позволяет совместить удобный ввод «как в чате» и строгий финансовый учет. Ключевое решение — отделить интерпретацию текста (ИИ) от расчетов и правил учета (Python-движок), а адаптивность обеспечить через управляемый контекст: снимки бюджета, стратегия, неопределенности и оценка реального эффекта рекомендаций. В качестве направления дальнейшей работы можно выделить подключение банковских выписок, расширение модели целей (несколько горизонтов) и внедрение детектора аномалий для раннего предупреждения о перерасходе.

Список литературы

1. OpenAI. Документация API: Structured Outputs, Function Calling, Compaction (дата обращения: 01.03.2026).
2. FastAPI Documentation. Building APIs with Python (дата обращения: 01.03.2026).
3. SQLite. Documentation. SQL Database Engine (дата обращения: 01.03.2026).
4. Thaler R. H. Mental accounting matters. Journal of Behavioral Decision Making, 1999.
5. OECD. Financial Literacy and Financial Inclusion: Results and Policies. Paris: OECD Publishing, 2018.