Приднестровский государственный университет им. Т.Г. Шевченко
Физико-технический институт

ОСНОВЫ ВЕБ-РАЗРАБОТКИ С ПРИМЕНЕНИЕМ ИИ-АССИСТЕНТОВ

Лабораторный практикум

Разработал:
ст. преподаватель
кафедры ИТ
Бричаг Д.В.

г. Тирасполь
2025 г.

Лабораторная работа №10

Мини-проект: «AI-powered веб-приложение»

Цель работы:

- Объединить все знания, полученные за курс: вёрстка, интерактивность, React, API, компоненты, состояние.
- Научиться использовать ИИ как **партнёра в разработке**, а не как замену мышлению.
- Создать **рабочее мини-веб-приложение** с реальным функционалом.
- Научиться отлаживать, интегрировать и улучшать сгенерированный код.

Выбор мини-проекта

Вы можете выбрать один из предложенных вариантов или предложить свой:

Вариант А — Новостная лента (News Feed)

Загрузка новостей через публичный API.

Приложение должно содержать минимальный функционал, состоящий из:

- Лента новостей
- Фильтры по категориям
- Кнопка “обновить”
- Пагинация или “загрузить ещё”
- Модальное окно для просмотра новости полностью

Вариант В — Галерея изображений с поиском

API: Unsplash, TheCatAPI, Pexels, Flickr.

Приложение должно содержать минимальный функционал, состоящий из:

- Поле ввода для поиска
- Грид картинок
- Ленивая загрузка (infinite scroll)
- Lightbox
- Сортировка / фильтры

Вариант C — Личный планировщик задач (Todo Pro)

Приложение должно содержать минимальный функционал, состоящий из:

- Списки задач
- Фильтры (сегодня / завтра / все)
- Локальное сохранение
- Мини-статистика (выполнено/осталось)
- Drag’n’drop задач (опционально)

Вариант B — Любой проект студента

Если он использует React + API + взаимодействие с ИИ.

Как итог, в этой работе вам нужно продемонстрировать понимание:

- компонентной архитектуры,
- маршрутов (если нужны),
- useEffect, useState, пропсов,
- работы с внешними API,
- UX-взаимодействий,
- принципов интеграции кода ИИ.

Практическая часть

Общие шаги, подходящие под любую выбранную тему в общем виде

Шаг 1. Подготовка окружения

Создать новый React-проект:

```
npm create vite@latest ai-webapp --template react
cd ai-webapp
npm install
npm run dev
```

Шаг 2. Архитектура проекта

Для приложения необходимо создать корректный «каркас». Для этого используем помощников, чтобы создать план:

- какие компоненты нужны,
- где хранится состояние,
- какие API понадобятся,
- как организовать UI.

ИИ может помочь составить архитектуру:

```
Сгенерируй архитектурный план React-приложения <тип приложения>.
Опиши структуру компонентов и состояние.
```

Шаг 3. Генерация основных компонентов через ИИ

Например:

```
Сделай компонент Gallery.jsx с поиском и загрузкой изображений через API Unsplash.
Используй useState и useEffect. Добавь loading и error.
```

Шаг 4. Создание дополнительного функционала вручную

Приложение должно содержать минимальный пользовательский функционал для удобного обращения с данными:

- модальное окно
- поиск
- фильтры
- сортировка
- сохранение истории
- кнопка «загрузить ещё»

Здесь ваша задача состоит в том, чтобы использовать ИИ для интеграции новых функциональных возможностей приложения по мере его роста и усложнения.

Шаг 5. Стилизация

Вы вольны использовать любой подход, который был рассмотрен в предыдущих лабораторных работах:

- Tailwind
- чистый CSS
- Bootstrap
- готовые UI-киты

Пример промпта для создания внешнего вида приложения:

```
Сгенерируй простой адаптивный CSS для галереи изображений.
Сетка 3 колонки, на мобильных — 1.
```

Шаг 6. Оптимизация и отладка

Необходимо аккуратно проверить, все возможности работы приложения. Убедитесь, что не возникают следующие проблемы:

- бесконечные циклы useEffect
- неожиданные повторные рендеры
- ошибки API
- пустые данные

- отсутствие ключей key в списках
- корректную работу UI

Работа с git и GitHub

1. Инициализируйте репозиторий:

```
git init
git add .
git commit -m "Lab9: React [theme] App"
```

2. Создайте новый репозиторий на GitHub.

3. Свяжите локальный проект с GitHub:

```
git remote add origin https://github.com/<user>/lab9-react-api.git
git branch -M main
git push -u origin main
```

Отчёт (в README.md)

Добавьте в файл README.md следующую информацию:

- Название выбранного проекта
- Скриншоты работы приложения
- Описание функционала
- Используемое API
- Примеры запросов
- Какие части сгенерировал ИИ
- Какие части написаны вручную
- Какие ошибки возникли при интеграции
- Как вы решили возникшие ошибки

Результаты работы

В итоге у вас должно быть:

1. Рабочее React-приложение.
2. Исходный код с компонентами и состоянием.
3. Реализован функционал согласно чек-листу методических материалов.
4. Репозиторий на GitHub и отчёт в README.md.

Критерии оценки

- Приложение запускается и работает (2 балла).
- Работает загрузка данных из API (2 балла).
- Работает загрузка новых данных по кнопке или скроллу (2 балла).
- Реализован функционал выбора / поиска / фильтрации (2 балла).
- Код выложен на GitHub и оформлен отчёт (2 балла).