

Возможность прогнозирования концентрации препарата и сравнения различных режимов дозирования до назначения.

Автоматическое документирование всех расчетов в базе данных.

Перспективы развития:

интеграция с лабораторными информационными системами;

пополнение базы лекарственных препаратов;

разработка мобильного приложения для медсестер;

применение методов машинного обучения для уточнения параметров модели.

Разработанное приложение может быть использовано в практическом здравоохранении для повышения качества фармакотерапии, а также в учебном процессе для демонстрации применения методов математического моделирования в медицине.

1 Лищук, В.А. Биомеханика: учебник для вузов / В.А. Лищук, Е.В. Москаленко. – Москва : Юрайт, 2023. – 412 с.

2 Кукес, В.Г. Клиническая фармакокинетика: теория и практика / В.Г. Кукес. – Москва : ГЭОТАР-Медиа, 2022. – 528 с.

3 SciPy User Guide [сайт]. – URL: : <https://docs.scipy.org/doc/scipy/> (дата доступа: 28.02.2026).

4 Документация PyQt5 [сайт]. – URL: <https://www.riverbankcomputing.com/static/Docs/PyQt5/> (дата доступа: 01.03.2026).

## ПРИМЕНЕНИЕ ГЕНЕРАТИВНЫХ ЯЗЫКОВЫХ МОДЕЛЕЙ ДЛЯ ГЕНЕРАЦИИ ПРОГРАММНОГО КОДА В ВЕБ-РАЗРАБОТКЕ

*Тужик Д.А.,*

*студент 4 курса ВГУ имени П.М. Машерова, г. Витебск, Республика Беларусь*

*Научный руководитель – Кухарев А.В., канд. физ.-мат. наук*

В последние годы наблюдается стремительное развитие технологий искусственного интеллекта, основанных на генеративных языковых моделях на архитектуре трансформеров [1, 2]. Такие модели способны анализировать большие объёмы текстовой информации и на основе выявленных закономерностей формировать новый текст, включая программный код. В современных исследованиях отмечается, что генеративные языковые модели демонстрируют хорошие результаты в задачах автоматической генерации программного кода, однако полученные решения требуют проверки со стороны разработчика [3, 4]. Актуальным является использование языковых моделей в веб-разработке, где значительная часть задач связана с созданием типовых элементов интерфейса и структуры веб-страниц.

Целью настоящей работы является исследование возможностей генеративных языковых моделей при генерации программного кода в веб-разработке.

**Материал и методы.** В качестве исследуемого инструмента использовалась языковая модель GPT-4o-mini компании OpenAI [5]. Эксперимент заключался в сравнении процесса разработки простых элементов веб-сайта при использовании генеративной модели и при традиционной ручной разработке.

Для проведения эксперимента были выбраны три типовые задачи веб-разработки:

– создание базовой HTML-структуры страницы;

– разработка CSS-стилей для оформления страницы;

– написание простого JavaScript-скрипта для обработки пользовательского действия (например, нажатия кнопки).

Каждая из указанных задач выполнялась три раза: один раз код создавался вручную и два раза — с использованием генеративной языковой модели ChatGPT. При использовании генеративной модели формулировался текстовый запрос, после чего полученный код запускался в браузере и при необходимости редактировался для устранения обнаруженных неточностей.

**Результаты и их обсуждение.** В ходе эксперимента было установлено, что языковая модель достаточно корректно формирует HTML-разметку и CSS-стили, особенно если задача сформулирована подробно и однозначно.

При проверке сгенерированного кода синтаксических ошибок обнаружено не было. Код успешно запускался и корректно выполнялся в браузере. Однако в отдельных случаях наблюдались небольшие логические неточности и несогласованность между различными частями кода. Было выявлено три типа проблем.

- 1) несогласованность между HTML и JavaScript;
- 2) неполная реализация функциональности обработчика события;
- 3) избыточность CSS-кода.

При ручной разработке для реализации рассматриваемых задач требовалось около 65–70 строк кода, включая HTML-разметку, CSS-стили и JavaScript-скрипт, а размер файлов составлял 2,1–2,4 килобайта. При использовании же генеративной языковой модели средний объём сгенерированного кода составлял 75–85 строк, а размер файлов 2,6–3,0 килобайта. Модель иногда генерирует дополнительные элементы разметки, комментарии или стили, которые не являются строго необходимыми для выполнения задачи. Сравнительные результаты представлены в таблице.

Таблица – Сравнение ручной разработки и разработки с помощью генеративной языковой модели GPT-4o-mini

Способ разработки	Среднее время выполнения задачи, мин.	Среднее количество строк кода на одну задачу	Суммарное количество исправлений
Ручное написание кода	17,8	68 строк	0
Использование GPT-4o-mini	6,9	81 строк	3

Применение генеративной языковой модели позволило уменьшить время выполнения задачи более чем в два раза. Это достигается за счёт автоматической генерации базовой структуры кода. При этом разработчик выполняет роль проверяющего и при необходимости вносит корректировки. Несколько больший объём сгенерированного кода по сравнению с кодом, написанным вручную, связан с тем, что модель стремится генерировать более универсальные и структурированные решения, которые могут включать дополнительные элементы разметки или стили. Однако такое увеличение объёма кода не оказало заметного влияния на производительность веб-страницы.

В большинстве случаев сгенерированный код имел понятную структуру и корректное форматирование. HTML-разметка была логично организована, а CSS-стили использовали распространённые способы оформления элементов. JavaScript-код также был написан с использованием стандартных методов языка.

**Заключение.** Проведённое исследование показывает, что генеративные языковые модели могут эффективно использоваться в качестве вспомогательного инструмента при разработке веб-сайтов. Использование языковых моделей архитектуры GPT позволяет значительно ускорить процесс написания кода, особенно при создании типовых элементов интерфейса и базовой структуры веб-страниц. В то же время такие модели пока не могут полностью заменить разработчика, поскольку сгенерированный код может содержать логические неточности или требовать дополнительной оптимизации. Поэтому генеративные языковые модели следует рассматривать прежде всего как инструмент поддержки программиста, позволяющий повысить эффективность разработки и сократить время выполнения стандартных задач.

1 Vaswani, A. Attention Is All You Need / A. Vaswani [et al.] // Advances in Neural Information Processing Systems. – 2017. – Vol. 30. – P. 5998–6008.

2 Brown, T. Language Models are Few-Shot Learners / T. Brown [et al.] // Advances in Neural Information Processing Systems. – 2020. – Vol. 33. – P. 1877–1901.

3 Research on the Application of Large Language Models in Code Generation [сайт] – URL: <https://arxiv.org/abs/2411.00932> (дата обращения: 04.03.2026)

4 Букунова, И. Н. Использование чат-ботов на основе нейросетей в программировании: обзор / И. Н. Букунова, М. Ж. Жасузакова, Б. А. Талпакова, Т. Э. Хусаинов // Вестник КазАТК. – 2023. – № 3 (126). – С. 222–231.

5 GPT-4o mini: advancing cost-efficient intelligence / OpenAI. – Электрон. текстовые дан. – Загл. с экрана. – URL: <https://openai.com/index/gpt-4o-mini-advancing-cost-efficient-intelligence/> (дата обращения: 04.03.2026).