

БОЛЬШИЕ ЯЗЫКОВЫЕ МОДЕЛИ В ОБУЧЕНИИ ВЕБ-РАЗРАБОТКЕ: ПРАКТИКА И ПЕДАГОГИЧЕСКИЕ ПРИНЦИПЫ

Нормуродов Дилмурод Гайратович

Ташкентский международный университет образования (TIUE), г. Ташкент, Узбекистан

E-mail: normurodovwork@gmail.com

DOI: 10.61587/mmit.tiue.uz.v1i1.441

Аннотация. В статье представлен практический опыт интеграции AI-ассистентов на основе больших языковых моделей (Claude, ChatGPT) в учебный процесс при преподавании курсов программной инженерии и веб-разработки на платформе Django в Ташкентском международном университете образования. Описаны два направления применения: создание преподавателем интерактивных HTML-учебных материалов и использование студентами LLM-инструментов на практических занятиях. Опыт получен в течение двух семестров (2024–2025 учебный год) при работе с двумя академическими группами (около 30 студентов). Сформулированы педагогические принципы интеграции AI-ассистентов в учебный процесс с учётом двуязычной (русско-узбекской) образовательной среды и региональной специфики высшей школы Узбекистана.

Ключевые слова: большие языковые модели, AI-ассистент, Claude, ChatGPT, Django, Python, интерактивные учебные материалы, цифровое образование, vibe coding, IT-образование Узбекистана.

USING AI ASSISTANTS IN EDUCATIONAL WEB APPLICATION DEVELOPMENT: EXPERIENCE OF INTEGRATING LLM TOOLS IN DJANGO AND PYTHON TEACHING

Normurodov Dilmurod Gayratovich

Tashkent International University of Education (TIUE), Tashkent, Uzbekistan

E-mail: normurodovwork@gmail.com

Abstract. This paper presents practical experience of integrating LLM-based AI assistants (Claude, ChatGPT) into the teaching process of software engineering and Django web development courses at Tashkent International University of Education. Two directions of application are described: instructor-driven creation of interactive HTML learning materials and student use of LLM tools during practical sessions. The experience was gained over two semesters (academic year 2024–2025) with two academic groups (approximately 30 students). Pedagogical principles for integrating AI assistants into the educational process are formulated, taking into account the bilingual (Russian–Uzbek) educational environment and the regional specifics of higher education in Uzbekistan.

Keywords: large language models, AI assistant, Claude, ChatGPT, Django, Python, interactive learning materials, digital education, vibe coding, IT education in Uzbekistan.

Введение

Стремительное развитие больших языковых моделей (Large Language Models, LLM) в 2023–2025 годах существенно изменило практику разработки программного обеспечения. Инструменты на основе LLM — Claude (Anthropic), ChatGPT (OpenAI), GitHub Copilot (Microsoft) — превратились из экспериментальных ассистентов в повседневные рабочие инструменты программистов. Эти изменения непосредственно затронули сферу IT-образования, поставив перед преподавателями ряд практических вопросов: как готовить учебные материалы в новых условиях, как организовать работу студентов с AI-ассистентами на занятиях, как сохранить глубину понимания при доступности готовых решений.

Особую актуальность данные вопросы приобретают в условиях высшей школы Узбекистана, где учебный процесс ведётся в двуязычной среде (русский и узбекский языки), а доступ к качественным локализованным учебным материалам по современным веб-технологиям ограничен. AI-ассистенты потенциально способны частично компенсировать этот дефицит, выступая в роли персонального тьютора для студента и инструмента быстрой подготовки локализованных материалов для преподавателя.

Цель настоящей статьи — описать и проанализировать практический опыт применения AI-ассистентов в учебном процессе на кафедре информационных технологий и математики Ташкентского международного университета образования (TIUE). В работе рассматриваются два направления: использование AI-инструментов преподавателем для создания интерактивных HTML-материалов и применение LLM-ассистентов студентами при выполнении практических заданий по Django и Python.

Обзор литературы

Применение LLM в обучении программированию активно исследуется международным научным сообществом. Каснечи и соавторы [1] в обзорной работе систематизируют возможности и риски использования больших языковых моделей в образовании, выделяя такие направления, как создание персонализированных учебных материалов, объяснение сложных концепций и поддержка написания кода, при этом отмечая риски академической недобросовестности и поверхностного усвоения знаний.

Прайтер и соавторы [2] в работе, посвящённой использованию GitHub Copilot студентами вводных курсов по программированию (CS1), наблюдали характерные паттерны взаимодействия. Авторы выделили как продуктивные сценарии (использование AI как «справочника»), так и проблемные (механическое принятие предложенного кода без понимания), что указывает на необходимость педагогического сопровождения.

Морадии и соавторы [3] в эмпирическом исследовании сравнили решения, генерируемые GitHub Copilot, с решениями человека и пришли к выводу: для опытных разработчиков Copilot является активом, но для начинающих программистов может представлять риск из-за того, что предлагаемый код может быть некорректным или сложным для понимания. Юлмаз и Юлмаз [4] исследовали восприятие ChatGPT студентами, обучающимися программированию: среди преимуществ опрошенные отметили скорость получения ответов, помощь в отладке и повышение уверенности; среди недостатков — формирование лени и неполные или некорректные ответы модели.

Концепция «vibe coding», получившая распространение в 2025 году [5], описывает подход, при котором разработчик формулирует задачу на естественном языке, а LLM генерирует реализацию. Применительно к образованию данный подход порождает педагогическую дилемму: он может быть продуктивным инструментом обучения при условии формирования у студента навыков критической оценки сгенерированного кода, либо же препятствовать формированию базовых программистских компетенций при некритическом использовании.

Метод исследования

Исследование носит характер описательного педагогического опыта, основанного на наблюдениях автора в роли преподавателя в течение двух семестров (осень 2024 — весна 2025). В исследование были вовлечены студенты двух академических групп бакалавриата по направлению «Программная инженерия» TIUE — около 30 студентов, изучающих курсы «Системы управления базами данных», «Программная инженерия» и связанные дисциплины с использованием Django и Python.

В качестве инструментальной базы использовались: Claude (Anthropic) — основной инструмент автора для подготовки интерактивных HTML-материалов; ChatGPT (OpenAI) и Claude — инструменты, доступные студентам на практических занятиях. Применялись следующие методы:

- педагогическое наблюдение за процессом выполнения практических заданий с использованием AI-ассистентов;
- анализ устных ответов студентов при защите лабораторных работ как индикатора глубины понимания написанного кода;
- фиксация формулировок промптов и итераций при создании интерактивных учебных материалов;
- экспертная оценка автором трудозатрат при подготовке материалов с использованием AI и без него.

Следует подчеркнуть, что приводимые в статье оценки трудозатрат носят характер субъективной экспертной оценки автора и не являются результатом формального хронометрического исследования. Их цель — дать качественное представление о порядке ускорения работы.

Результаты и обсуждение

Создание интерактивных HTML-материалов с помощью AI

Одним из наиболее продуктивных применений LLM в практике автора оказалось использование Claude для генерации интерактивных HTML-лекций. Типовой рабочий процесс включает следующие этапы: формулирование развёрнутого промпта с указанием темы, уровня аудитории, требуемых разделов и интерактивных элементов; получение от модели полноценного HTML-файла со встроенным CSS и JavaScript; ручная редакция примеров под учебный контекст; распространение готового файла среди студентов. Важной особенностью является то, что результат открывается в любом браузере без установки дополнительного программного обеспечения, что критично в условиях вычислительных классов с разной программной средой.

В качестве примера приведём фрагмент промпта, использованного автором для генерации лекции по теме «Аутентификация в Django»:

Создай интерактивный HTML-урок по аутентификации в Django для студентов 3 курса бакалавриата (программная инженерия).

Включи следующие разделы:

- обзор встроенной системы User в Django,
- регистрация и вход с использованием UserCreationForm,
- защита представлений через @login_required,
- работа с сессиями и кастомные модели User.

Добавь подсветку синтаксиса, мини-тесты после каждого раздела и поясняющие диаграммы. Тёмная тема. Язык — русский.

Полученный HTML-файл потребовал лишь точечной редакции примеров кода для приведения к учебному контексту курса (использование уже знакомых студентам моделей данных). Аналогичным образом были созданы интерактивные лекции по Big Data на Python (pandas, numpy, matplotlib), реляционным базам данных и SQL, разработке REST API на Django REST Framework. Каждый материал содержит объяснение концепций, примеры кода, наглядные диаграммы и контрольные вопросы для

самопроверки. Фрагмент сгенерированной интерактивной лекции по аутентификации в Django представлен на Рисунке 1.

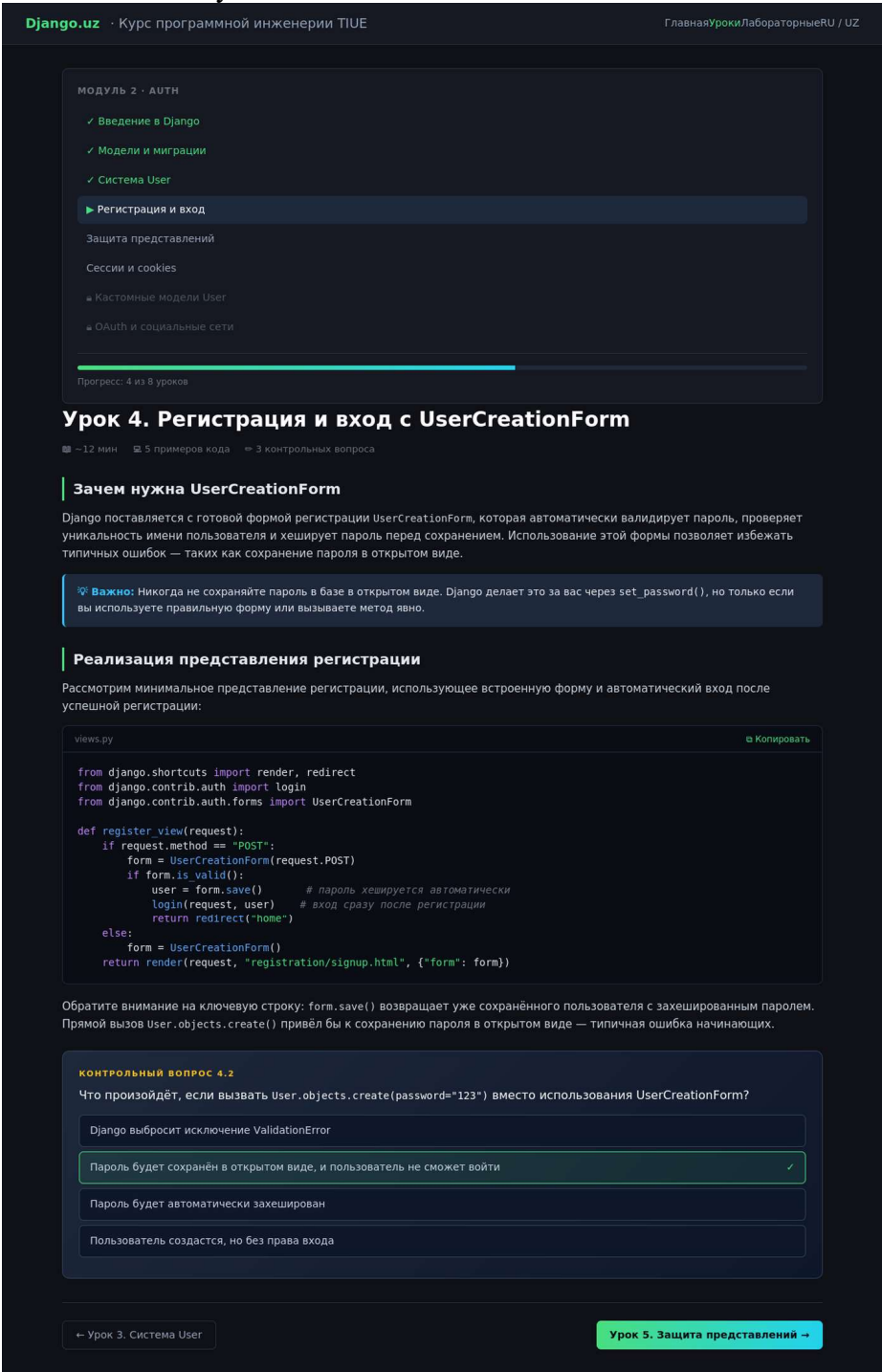


Рисунок 1. Фрагмент интерактивной HTML-лекции по аутентификации в Django, сгенерированной с помощью Claude (тёмная тема, подсветка синтаксиса, встроенный контрольный вопрос)

Сравнительная экспертная оценка трудозатрат автора при подготовке учебных материалов представлена в Таблице 1.

Таблица 1. Сравнительная экспертная оценка трудозатрат на подготовку учебных материалов

Тема лекции	Без AI (часы, оценка)	С AI-ассистентом (часы, оценка)
-------------	-----------------------	---------------------------------

Реляционные БД и SQL	≈ 5–6	≈ 1.0–1.5
Аутентификация в Django	≈ 4–5	≈ 0.5–1.0
Big Data с Python	≈ 6–8	≈ 1.5–2.0
Django REST API	≈ 5–7	≈ 1.0–1.5

Принципиальное преимущество подхода в условиях двуязычной образовательной среды Узбекистана состоит в возможности оперативно генерировать материалы как на русском, так и на узбекском языке. Это особенно актуально для студентов из регионов, у которых уровень владения русским языком может быть ниже, чем у столичных студентов.

Применение AI-ассистентов студентами при написании кода

На практических занятиях по Django студентам было разрешено использовать AI-ассистенты (ChatGPT, Claude) при выполнении лабораторных работ. По итогам наблюдения автор выделил два характерных паттерна взаимодействия — продуктивный и проблемный.

Продуктивный паттерн проявлялся в формулировании конкретных вопросов («Объясни, почему возникает ошибка IntegrityError при сохранении модели заказа»), в использовании AI для понимания незнакомых конструкций Django ORM и в запросе альтернативных вариантов реализации для сравнения подходов. Подобное взаимодействие приближается к работе с интерактивным учебником и согласуется с наблюдениями Юлмаза и Юлмаз [4] о ChatGPT как ускорителе процесса обучения.

Проблемный паттерн выражался в попытках вставить сгенерированный код без его понимания, в неспособности студента объяснить написанное при защите лабораторной работы и в затруднениях при отладке нетипичных ошибок. Аналогичные наблюдения описаны Прайтером и соавторами [2] для пользователей GitHub Copilot.

Иллюстративный кейс. На занятии по Django ORM студент получил от ChatGPT код, использующий метод `select_related()` для оптимизации запросов. Студент скопировал решение, и оно работало корректно. На защите лабораторной работы студенту был задан вопрос: «Почему здесь использован `select_related`, а не `prefetch_related`?». Студент не смог обосновать выбор. Был организован разбор: что такое JOIN на уровне SQL, какой запрос порождают оба метода, когда применим каждый из них. После разбора студент самостоятельно выполнил аналогичное задание уже без обращения к AI. Этот кейс показывает, что обязательная защита кода превращает AI-ассистента из «решателя задач» в стартовую точку для формирования понимания.

В Таблице 2 обобщены наблюдаемые преимущества и риски использования AI-ассистентов в учебном процессе.

Таблица 2. Преимущества и риски использования AI-ассистентов в IT-обучении

Преимущества	Риски
Существенное сокращение трудозатрат на создание учебных материалов	Поверхностное понимание кода при некритическом копировании
Повышение интерактивности и визуального качества лекций	Снижение навыков самостоятельной отладки
Объяснения на родном языке студента (русский, узбекский)	Риск академической недобросовестности при выполнении заданий
Формирование навыков формулирования задач (промт-инжиниринг)	Зависимость от инструмента в ущерб базовым компетенциям
Доступность 24/7 как персонального консультанта	Возможные ошибки и «галлюцинации» модели

Педагогические принципы интеграции LLM в учебный процесс

На основе двухсеместрового опыта автором сформулированы следующие педагогические принципы интеграции AI-ассистентов в IT-обучение:

– Принцип обязательной защиты кода: любой код, выполненный с участием AI, студент должен уметь объяснить и обосновать в устной форме.

– «AI как справочник, а не как решебник»: инструмент применяется для понимания незнакомых концепций, а не для получения готового решения «под ключ».

– Демонстрационная итерация: преподаватель в реальном времени демонстрирует процесс работы с AI — формулирование промпта, оценку ответа, дополнительные уточнения. Это формирует у студента модель «правильного» взаимодействия.

– Контрольные срезы без AI: письменные контрольные работы и экзамены проводятся без использования AI-инструментов, обеспечивая объективную оценку сформированных знаний.

– Учёт двуязычной среды: AI-ассистенты применяются как инструмент локализации учебных материалов, что особенно ценно в условиях ограниченного объёма качественных русско- и узбекоязычных ресурсов по современным веб-технологиям.

Применение данных принципов в течение двух семестров позволило существенно снизить частоту проблемных паттернов использования AI и одновременно сохранить положительный эффект ускорения учебного процесса.

Заключение

Опыт интеграции AI-ассистентов в преподавание Django и Python на кафедре информационных технологий и математики TIUE позволяет сформулировать следующие выводы. Использование Claude и ChatGPT для подготовки интерактивных HTML-учебных материалов существенно сокращает трудозатраты преподавателя при сохранении и даже повышении качества контента. Применение AI-ассистентов студентами эффективно при работе в режиме «справочника» и пояснителя, однако требует педагогических механизмов контроля понимания — в первую очередь, обязательной устной защиты кода.

Ключевой вывод состоит в том, что LLM-инструменты не заменяют, а трансформируют роли участников образовательного процесса. Преподаватель смещается от транслятора знаний к архитектору учебного опыта; студент — от пассивного получателя информации к активному участнику, владеющему навыками формулирования задач и критической оценки сгенерированных решений.

Для условий высшей школы Узбекистана отдельную ценность представляет возможность оперативной локализации учебных материалов на русский и узбекский языки, что частично компенсирует дефицит качественных локализованных ресурсов по современным IT-технологиям. Перспективы дальнейших исследований автор связывает с разработкой формализованных методик оценки компетенций промпт-инжиниринга как самостоятельного учебного результата и с проведением количественного исследования влияния интеграции AI-ассистентов на успеваемость студентов в условиях узбекистанской высшей школы.

Список литературы

1. Kasneci, E., Sessler, K., Küchemann, S., Bannert, M., Dementieva, D., Fischer, F., et al. (2023). ChatGPT for good? On opportunities and challenges of large language models for education. *Learning and Individual Differences*, 103, 102274. <https://doi.org/10.1016/j.lindif.2023.102274>
2. Prather, J., Reeves, B. N., Denny, P., Becker, B. A., Leinonen, J., Luxton-Reilly, A., Powell, G., Finnie-Ansley, J., & Santos, E. A. (2023). "It's Weird That It Knows What I Want": Usability and Interactions with Copilot for Novice Programmers. *ACM Transactions on Computer-Human Interaction*, 31(1), Article 4. <https://doi.org/10.1145/3617367>
3. Moradi Dakhel, A., Majdinasab, V., Nikanjam, A., Khomh, F., Desmarais, M. C., & Jiang, Z. M. J. (2023). GitHub Copilot AI pair programmer: Asset or liability? *Journal of Systems and Software*, 203, 111734. <https://doi.org/10.1016/j.jss.2023.111734>

4. Yilmaz, R., & Karaoglan Yilmaz, F. G. (2023). Augmented intelligence in programming learning: Examining student views on the use of ChatGPT for programming learning. *Computers in Human Behavior: Artificial Humans*, 1(2), 100005. <https://doi.org/10.1016/j.chbah.2023.100005>
5. Karpathy, A. (2025, February 3). Vibe coding [Post on X]. <https://x.com/karpathy/status/1886192184808149190>
6. Sallam, M. (2023). ChatGPT Utility in Healthcare Education, Research, and Practice: Systematic Review on the Promising Perspectives and Valid Concerns. *Healthcare*, 11(6), 887. <https://doi.org/10.3390/healthcare11060887>
7. Becker, B. A., Denny, P., Finnie-Ansley, J., Luxton-Reilly, A., Prather, J., & Santos, E. A. (2023). Programming Is Hard — Or at Least It Used to Be: Educational Opportunities and Challenges of AI Code Generation. *Proceedings of the 54th ACM Technical Symposium on Computer Science Education V.1 (SIGCSE 2023)*, 500–506. <https://doi.org/10.1145/3545945.3569759>
8. Denny, P., Kumar, V., & Giacaman, N. (2023). Conversing with Copilot: Exploring Prompt Engineering for Solving CS1 Problems Using Natural Language. *Proceedings of the 54th ACM Technical Symposium on Computer Science Education V.1 (SIGCSE 2023)*, 1136–1142. <https://doi.org/10.1145/3545945.3569823>